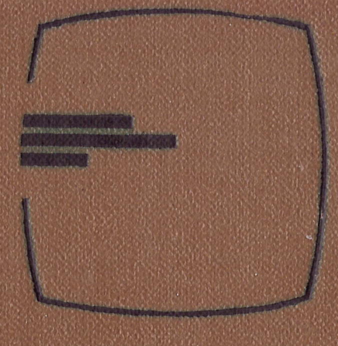
РАБОТАЕМ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

РОБОТРОН 1715



РАБОТАЕМ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ РОБОТРОН 1715

Под общей редакцией
канд. техн. наук **Н. В. Макаровой**



Ленинград
"Машиностроение"
Ленинградское отделение
1989

ББК 32.973.2
Р13
УДК 519.6

Авторы: Н. В. Макарова, А. А. Докучаев, В. Н. Египко, Д. Н. Семенов

Рецензент канд. техн. наук проф. В. В. Кириллов

Работаем на персональном компьютере Робот-
Р13 рон 1715/Н. В. Макарова, А. А. Докучаев, В. Н. Египко, Д. Н. Семенов; Под общ. ред. канд. техн. наук Н. В. Макаровой. — Л.: Машиностроение. Ленингр. отд-ние, 1989. — 367 с.: ил.

ISBN 5-217-00703-6

Основная цель книги — ознакомить пользователя со структурой и принципом работы современного персонального компьютера Роботрон 1715, обучить работе на нем, привив навыки обращения с пакетами прикладных программ: текстовым редактором (РЕФОР), электронной таблицей (ВАРИТАБ), картотекой (КАРТ1715), системой управления реляционной базой данных (РЕБУС или dBASE). Кроме того, изложены сведения по алгоритмическому языку Бейсик.

Книга предназначена для широкого круга пользователей ЭВМ, желающих освоить компьютер Роботрон 1715, а также может быть использована программистами как пособие по ускоренному его освоению.

Р $\frac{2404040000-966}{038(01)-89}$ КБ—53—22—88 ББК 32.973.2

ISBN 5-217-00703-6

ПРЕДИСЛОВИЕ

Обработка информации в различных ее формах, видах, представлениях воспринимается как обычная производственная деятельность. В промышленно развитых странах обработка информации — одна из самых существенных сфер человеческой деятельности. Так, в США на информационную работу во временно́м и стоимостном выражении приходится более 2/3 всех трудовых затрат.

Сравнительно недавно появилось понятие, отражающее современный подход к деятельности людей по переработке информации,— информационная технология. Под информационной технологией понимается совокупность процессов сбора, передачи, обработки, хранения информации, выдачи результатов потребителю с использованием компьютерной техники. Что существенно нового можно заметить в этом понятии? Несомненно, сбором, обработкой, хранением информации люди занимались на протяжении всей истории развития общества. Но появление и внедрение компьютеров заставило пересмотреть стереотипы обработки информации и процессов, происходящих в любой сфере человеческой деятельности.

С каждым годом компьютер все настойчивее заявляет о себе. Уже невозможно не заметить, что это техническое средство при умелом его использовании может не только выполнять рутинные операции, как в первые периоды его применения, но и оказывать реальную помощь в принятии решения, облегчить ваш труд, освободить время для творческой работы. Этот процесс еще больше активизировался

и приобрел необратимый характер с появлением персонального компьютера и осознания его возможностей. Сегодняшнее время, по-видимому, можно оценивать как этап зарождения информационной технологии. Еще во многом неясны концепции информационной технологии, специалисты очень осторожно прогнозируют ее будущее, но одно, несомненно, ясно — мы должны овладеть методикой работы с персональным компьютером как основой всей современной и будущей информационной технологии.

Освоить компьютерную грамоту вы должны в такой степени, чтобы не возникало чувство робости перед незнакомой техникой, чтобы компьютер стал таким же привычным средством работы, как карандаш, ручка, бумага, линейка. И это вполне реально, так как и дизайн компьютера, и комплекс программных средств по организации обработки информации ориентированы на организацию «дружественной» среды общения человека и компьютера.

Подобные комплексы программных средств, называемые пакетами прикладных программ, призваны реализовать конкретные цели информационной технологии, а именно: автоматизировать процесс подготовки текстов, исключив тем самым машинописные работы и многие традиционно сложившиеся элементы делопроизводства; изменить процедуру подготовки документов в виде таблиц, что позволит сократить отводимые для этого время и средства; организовать хранение больших объемов информации; обеспечить простоту и удобство доступа к информации и корректировки ее и т. д. Работа на персональном компьютере пользователя, не являющегося специалистом в области компьютерной техники, заключается в свободном владении им средствами входного языка пакета прикладных программ.

Одним из первых современных персональных компьютеров, получившим широкое распространение в нашей стране, является Роботрон 1715 производства ГДР. Он имеет традиционный состав устройств как по внешнему

оформлению, так и по концепции их использования в соответствии с принятым стандартом для персональных компьютеров. Хотя Роботрон 1715 и относится к группе компьютеров, в структуре которых применен 8-разрядный микропроцессор, но в перспективе более мощные его модели будут построены на базе 16-разрядных микропроцессоров. Этот компьютер современного направления уверенно может быть включен в общую систему информационной технологии.

В настоящей книге вы ознакомитесь с основными принципами работы в среде пакетов прикладных программ. Применительно к каждому из пакетов прикладных программ изложение материала ведется по одной и той же схеме. Вначале рассмотрены назначение пакета и его возможности, затем на конкретной задаче по методу KEY-BY-KEY (клавиша за клавишей) показано, как использовать входной язык пакета для ее решения. Настоятельно рекомендуем вам все действия, связанные с решением конкретной задачи, проделать самостоятельно и в той последовательности, как изложено в книге. Только так можно достаточно просто и быстро понять стратегию работы в среде конкретного пакета прикладных программ. Основные возможности пакета вы освоите на конкретной задаче. Дополнительные же средства работы в среде пакета представлены в виде справочной информации в конце каждой главы.

В книге рассмотрены четыре пакета прикладных программ, используемых в персональном компьютере Роботрон 1715, и алгоритмический язык Бейсик. Первые три пакета — текстовый редактор РЕФОР, электронная таблица ВАРИТАБ, система ведения картотеки КАРТ1715 — рассчитаны на пользователя, знающего только основы компьютерной грамоты [6]. Пакет системы управления реляционной базой данных РЕБУС, являющийся своего рода системой программирования, как и язык Бейсик, ориентирован, скорее, на профессиональных программистов.

Материал, изложенный в гл. 1 и 2, является базой, без знания которой невозможна ваша работа на компьютере. В гл. 1 описаны аппаратная часть компьютера, назначение и роль каждого устройства, показано, как начать работу и управлять ею посредством клавиатуры. Глава 2 познакомит вас с операционной системой и обслуживающими командами.

Глава 3 научит вас работать с текстовым редактором по подготовке, редактированию и печати текстов различного содержания и оформления.

Глава 4 полезна тем, кто часто имеет дело с информацией, представленной в табличной форме. Пакет прикладных программ, предназначенный для работы с электронной таблицей, позволит быстро создавать и печатать такие таблицы, проводить в них расчеты.

В гл. 5 вы получите сведения, как организовать хранение информации не в картотеке, а в компьютере, тем самым сократив время поиска информации и освободив пространство, занимаемое картотекой на бумажном носителе.

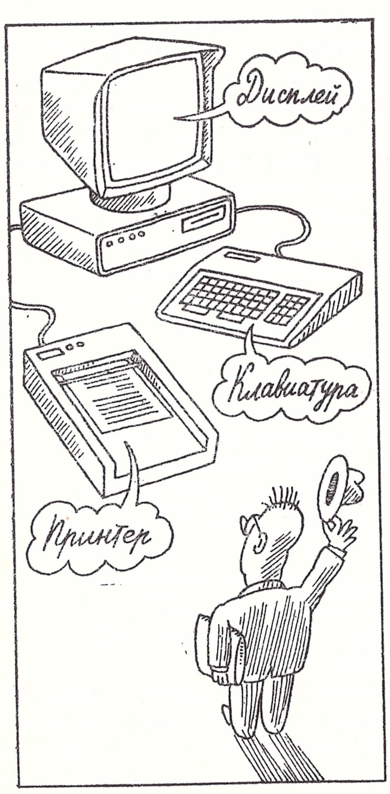
Глава 6 представит интерес для программистов, желающих ознакомиться с концепцией работы системы управления реляционной базой данных. Показано, как вносить данные в базу, проводить их коррекцию, извлекать данные из базы в режиме ввода одноразовых команд и режиме работы командного файла.

В гл. 7 изложены базовые конструкции алгоритмического языка Бейсик на конкретных примерах. Ознакомившись с ними, вы научитесь составлять несложные программы.

Авторы надеются, что, неукоснительно соблюдая методику решения предлагаемых задач на компьютере Роботрон 1715, вы быстро освоите стратегию работы в среде рассматриваемых пакетов и, получив столь мощное инструментальное средство, найдете ему применение в вашей деятельности. Желаем успехов!

1

ЗНАКОМИМСЯ С АППАРАТНОЙ ЧАСТЬЮ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА РОБОТРОН 1715



1.1. ОСОБЕННОСТИ СТРУКТУРНОЙ СХЕМЫ РОБОТРОНА 1715

Что представляет собой традиционная структурная схема персонального компьютера?

Порадуемся вместе с вами: вы являетесь обладателем персонального компьютера Роботрон 1715. Перед вами на столе стоят электронные блоки, на которые вы смотрите с некоторым почтением и надеждой, а возможно даже, в душе возникает сомнение: справлюсь ли я, сумею ли освоить их. Не надо сомневаться, настройтесь на то, что пользоваться персональным компьютером вскоре станет для вас так же привычно, как телевизором

или магнитофоном. Вас ведь вряд ли интересует принцип работы схем телевизора и магнитофона, так и здесь вам не потребуется знаний глубин микроэлектроники и происходящих процессов. Однако понимание того, зачем нужны определенные устройства, какие функции на них возлагаются и какая роль им отводится, является для вас необходимостью и залогом будущих успехов. Хотя, надеемся, вы освоили компьютерную грамоту [6], краткое повторение ее основ поможет вам быстрее разобраться в назначении устройств персонального компьютера Роботрон 1715.

Совокупность электронных блоков, из которых состоит компьютер, называют аппаратной частью или комплексом технических средств. Работа аппаратной части компьютера определяется программой. Пограмма представляет собой описание процесса обработки или управления на символическом языке, воспринимаемом компьютером. Без программы аппаратная часть — это груда бесполезных электронных и механических блоков. Аппаратную часть условно можно представить в виде центральной и периферийной частей (рис. 1.1).

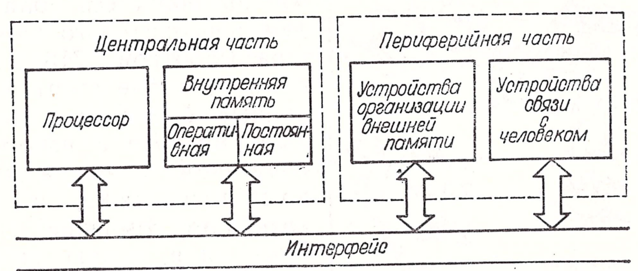


Рис. 1.1. Типовая структурная схема персонального компьютера

Центральную часть составляют процессор и внутренняя память (оперативная и постоянная). Процессор выполняет арифметические и логические действия, а также управляет работой основных узлов и блоков компьютера.

Память любого принципа действия и исполнения предназначена для хранения информации и программ.

●●● Оперативная память хранит текущую информацию и текущую программу решения задачи в данный момент времени, а также ряд вспомогательных программ для организации работы компьютера. В оперативную память можно записывать данные и программы и читать их из нее. Этот процесс схож с происходящим во время занятий: преподаватель записывает мелом на доске ту или иную информацию, а учащиеся читают ее с доски. После занятий доску очищают от мела. Так и в компьютере — при отключении питания содержимое оперативной памяти стирается.

●●● Постоянная память хранит вспомогательные программы, неоднократно используемые в процессе решения любых задач. К постоянной памяти можно обращаться только для чтения информации и программ, записать в нее новую их порцию невозможно, так как в постоянную память запись осуществлена в процессе изготовления компьютера на заводе. Постоянную память можно сравнить с книгой, которая была однажды создана, но пользуются ею многократно.

Одна из основных характеристик памяти, роль которой вы должны четко осознать,— это ее емкость. Емкость памяти характеризует максимальный объем информации, хранимой в ней, и выражается в килобайтах (Кбайт), мегабайтах (Мбайт), гигобайтах (Гбайт).

Напомним!

1 байт=8 бит (8 двоичных разрядов). В 1 байте можно хранить один символ, букву.

1 Кбайт≈1000 байт.

1 Мбайт≈1000 Кбайт=10^6 байт.

1 Гбайт≈1000 Мбайт=10^9 байт.

●●● Периферийную часть любого компьютера составляют устройства связи человека и компьютера (дисплей, клавиатура, печатающее устройство, графопостроитель) и устройство организации хранения информации во внеш-

ней памяти (дисководы, накопители на магнитной ленте). В дальнейшем мы предоставим вам возможность ознакомиться с этими устройствами подробнее. А сейчас лишь напомним, что техническую связь и взаимодействие всех устройств между собой осуществляет интерфейс — системная шина. Теперь можно перейти к рассмотрению аппаратной части персонального компьютера Роботрон 1715.

Состав аппаратной части Роботрона 1715

Структурная схема персонального компьютера Роботрон 1715 аналогична вышеописанной (рис. 1.1), однако при ее конструктивном оформлении не было необходимости отделять центральную часть от периферийной. В настоящее время определен стандарт на конструкцию персонального компьютера в виде автономных блоков: системного блока, дисплея, клавиатуры, печатающего устройства (принтера) (рис. 1.2).

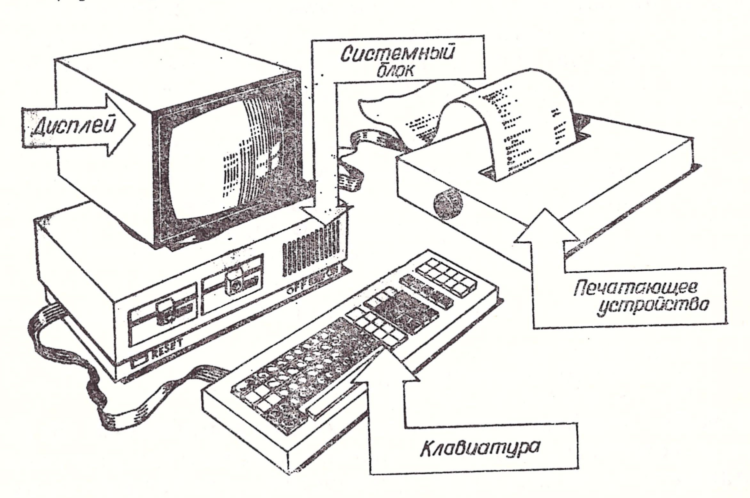


Рис. 1.2. Роботрон 1715

Системный блок является центром и основой персонального компьютера. Здесь сосредоточена вся его мощь, позволяющая реально воплотить замыслы человека относительно способа обработки информации.

Дисплей, клавиатура, печатающее устройство — это вспомогательная часть, предоставляющая пользователю возможность общения с компьютером. Как глаза и уши позволяют человеку воспринимать (вводить) информацию, так и клавиатура организует восприятие информации компьютером. Речевая деятельность человека (вывод информации) служит средством общения его с внешним миром. В компьютере вывод информации и восприятие ее в удобной для пользователя форме обеспечиваются печатающим устройством и дисплеем.

На рис. 1.2 показан общий вид Роботрона 1715. На системном блоке установлен дисплей. Перед системным блоком располагается клавиатура, которую легко установить в удобном для вас положении. Рядом вы видите печатающее устройство. Эти устройства имеют небольшие габаритные размеры и массу, что позволит вам разместить их на рабочем столе и установить в удобном для пользования ими положении. Внешнее оформление аппаратной части соответствует современным требованиям дизайна.

1.2. СОСТАВ СИСТЕМНОГО БЛОКА

Назначение

Устройства, входящие в системный блок, организуют обработку и хранение информации, управляют работой всех узлов и блоков персонального компьютера. Из его назначения вам уже ясно, что системный блок имеет сложную структуру. Но нет надобности разбираться в тонкостях его работы. Это удел специалистов. Вам достаточно только ознакомиться с назначением его основных со-

ставляющих: системной платы, дисководов, адаптерных плат, источника питания.

Системная плата

Системная плата — центральная электронная часть не только системного блока, но и всего персонального компьютера. На ней установлены интегральные схемы различного назначения. Интегральной схемой называется электронная схема, выполненная в виде кристалла кремния площадью 25—40 мм². Характеризует интегральную схему степень интеграции — количество первичных элементов (диодов, транзисторов, резисторов, конденсаторов), расположенных на одном кристалле. Благодаря специальной технологии обеспечивается степень интеграции от нескольких сотен до десятков тысяч первичных элементов. Интегральная схема размещена в герметичном стандартном корпусе с металлическими выводами, предназначенными для подачи входных воздействий и снятия выходных сигналов. Рассмотрим назначение основных интегральных схем системной платы.

●●● Переработку информации и управление всем процессом обработки осуществляет микропроцессор, т. е. процессор, выполненный в виде интегральной схемы. Микропроцессор содержит несколько десятков тысяч первичных элементов. В Роботроне 1715 используется 8-разрядный микропроцессор U880, аналогичный Z80 (фирмы «Зайлог»). Его структура ориентирована на обработку информации с длиной слова, равной 8 двоичным разрядам (битам). При сравнении 8-разрядного микропроцессора с широко распространенными в других персональных компьютерах 16-разрядными выявляются существенные недостатки: меньше скорость работы и существенно меньше объем оперативной памяти.

●●● На системной плате имеется ряд интегральных схем для организации оперативной и постоянной памяти. Объем оперативной памяти составляет 64 Кбайт. В по-

стоянной памяти обязательно должна храниться программа автоматической загрузки в оперативную память операционной системы.

●●● Остальные интегральные схемы играют вспомогательную роль при организации связи между устройствами.

Дисководы

По команде центрального процессора дисковод производит поиск и чтение информации с диска, а также запись информации на диск. Вы можете встретить также другое название дисковода — накопитель на гибком или жестком магнитном диске. До появления персонального компьютера дисковод входил в комплект ЭВМ как автономная конструкция, занимающая определенную площадь, и устанавливали его отдельно от других устройств. Дальнейшее развитие интегральных схем со все возрастающей степенью интеграции, а особенно появление микропроцессора позволили существенно уменьшить габариты центральной части и свести по сути всю центральную часть к одной системной плате. Это послужило толчком к интенсификации процесса, направленного на уменьшение габаритов

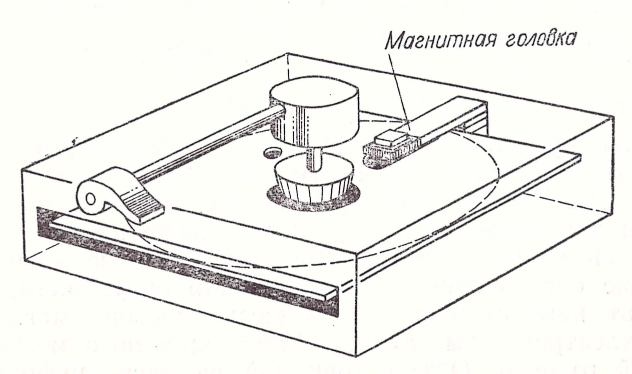


Рис. 1.3. Дисковод с гибким диском

внешних устройств, и к разработке малогабаритных дисководов.

Роботрон 1715 имеет два дисковода для гибких магнитных дисков (флоппи-дисков или дискетов). Упрощенное представление о принципах работы дисковода можно получить на примере использования бытового проигрывателя, где звучание достигается за счет соприкосновения головки воспроизведения с дорожкой вращающейся пластинки. Гибкий диск через специальное отверстие вставляется в дисковод и закрепляется на оси двигателя (рис. 1.3). Магнитная головка производит запись и считывание информации с поверхности вращающего диска. Роботрон 1715 может быть укомплектован дисководами разного типа.

●●● MFS 1.2 имеет одну головку и обеспечивает запись информации на 40 дорожек и считывание ее с одинарной плотностью.

●●● MFS 1.4 имеет одну головку и обеспечивает запись информации на 80 дорожек и считывание ее с двойной плотностью.

●●● MFS 1.6 имеет две головки и обеспечивает запись информации с двойной плотностью на каждую поверхность диска по 80 дорожек и считывание ее.

Замечание. Под плотностью здесь понимается количество дорожек, размещенных на диске (радиальная плотность).

Как устроен гибкий диск (дискета)?

Диск представляет собой гибкую пластину, покрытую магнитной пленкой; помещен он в пластиковый чехол (рис. 1.4). Так же, как и грампластинка, диск имеет дорожки для хранения информации, однако на поверхности диска вы их не заметите. Дорожки — это не физически, реально обозначенные на поверхности окружности, а результат намагничивания отдельных частиц магнитного слоя электрическим током, протекающим по обмотке магнитной головки. Объем хранимой на диске информации

выражается в килобайтах. В Роботроне 1715 используются диски максимальной емкости 184, 384 и 770 Кбайт в зависимости от типа дисковода.

Можно ли представить это себе в более наглядной форме и тем самым решить, хватит ли данного объема для реализации конкретной задачи? Приведем небольшой расчет. На машинописной странице помещаются 1800 симво-

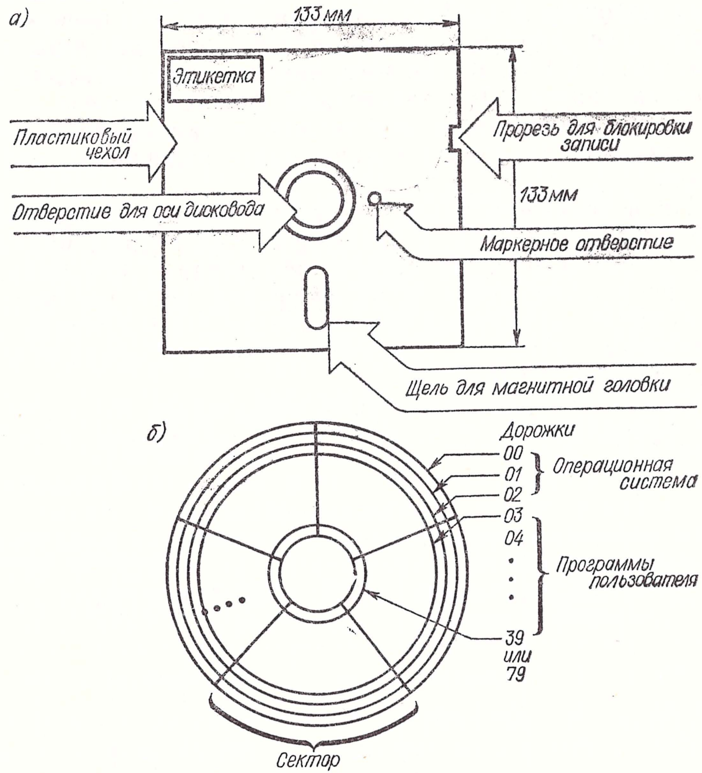


Рис. 1.4. Гибкий диск: *а* — в чехле; *б* — структура диска

лов из расчета 30 строк по 60 символов в каждой. Для хранения на диске одного символа потребуется 1 байт, одной страницы машинописного текста — 1800 байт ≈ ≈ 1,8 Кбайт, 100 страниц машинописного текста — 180 Кбайт, 200 страниц машинописного текста — 360 Кбайт. Сопоставьте эти цифры с емкостью дисков!

Внешнее оформление диска стандартное. На пластиковом чехле имеются маркерное отверстие для указания начала дорожки и прорезь, которая открывает поверхность диска для контакта с магнитной головкой дисковода (см. рис. 1.3, 1.4). Кроме того, предусмотрен вырез для блокировки записи. При заклеенном фольгой вырезе запись на дискету произвести невозможно и вам предоставляется возможность только чтения ранее записанной информации.

На поверхность чехла наклеена этикетка фирмы, производящей эти диски, с указанием стандартных параметров ранжирования по степени возрастания его емкости. Условное обозначение и расшифровка этих параметров, а также рекомендуемый тип дисковода для Роботрона 1715 приведены в табл. 1.1. Наиболее ограничена сфера использования диска SS/SD.

Запись на диск может быть произведена только после его форматирования, т. е. задания определенной струк-

Таблица 1.1

Типы дисков

Условное обозначение	Количество рабочих поверхностей	Плотность	Число дорожек	Емкость, Кбайт	Применение дисковода		
					MFS 1.2	MFS 1.4	MFS1.6
SS/SD	1	Одинарная	40	184	+	—	—
SS/DD	1	Двойная	80	380	+	+	—
DS/DD	2	„	2×80	770	+	+	+

туры. Сделать это можно командой INIT, подробно описанной в параграфе 2.3. В результате форматирования в зависимости от плотности записи диск будет содержать 40 или 80 дорожек. Пронумерованы дорожки от 0 до 39 (79) начиная с внешнего края (см. рис. 1.4). Каждая дорожка разбита на пять секторов емкостью по 1024 байта. Дорожки с номерами 0, 1, 2 всегда отводятся для операционной системы. Часть третьей дорожки занимает каталог диска. Остальные дорожки используются для хранения программ пользователя, данных, пакетов прикладных программ.

Адаптерная плата

Адаптерная плата (адаптер) — это составная часть интерфейса для соединения периферийных устройств с процессором. Интерфейс персонального компьютера технически реализован в виде адаптера, системной шины и разъемов. Для каждого периферийного устройства используют обычно свою адаптерную плату, однако существуют адаптеры, объединяющие несколько функций. Адаптерная плата содержит несколько интегральных схем, составляющих контроллер для управления периферийным устройством определенного типа. Работа адаптера осуществляется под управлением специальной программы — драйвера. Драйвер и контроллер должны соответствовать друг другу. Заменяя контроллер и драйвер, можно использовать адаптерную плату для подключения других периферийных устройств. Адаптерная плата подсоединяется к системной шине, представляющей собой провода для передачи информационных и управляющих сигналов и разъемы.

Источник питания

Источник питания подает напряжение соответствующего номинала на внутренние электронные схемы. Он подключается переводом переключателя, находящегося на

передней панели системного блока с правой стороны, в положение **ON** (см. рис. 1.2). Для предупреждения нагрева электронной части используется вентилятор.

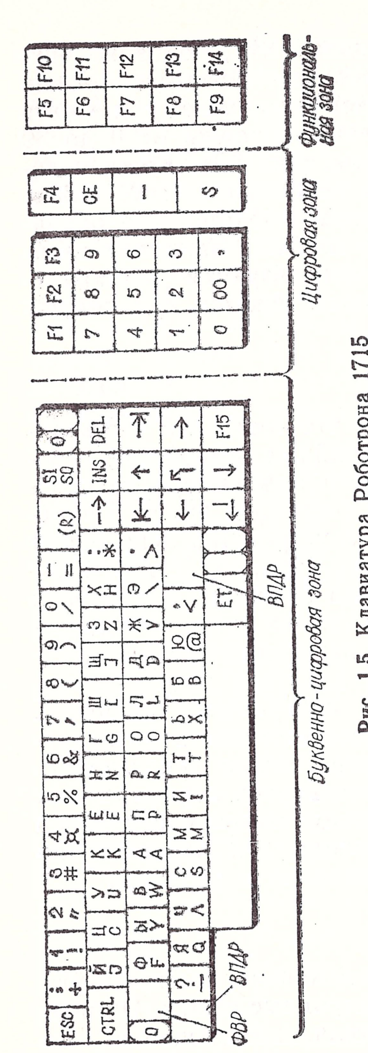


Рис. 1.5. Клавиатура Роботрона 1715

1.3. КАК ПОСРЕДСТВОМ КЛАВИАТУРЫ И ДИСПЛЕЯ ОРГАНИЗОВАТЬ РАБОТУ?

Зачем нужны клавиатура и дисплей?

С помощью клавиатуры вы вводите команды, числа, текст, программы в компьютер. Для проверки правильности ввода следует воспользоваться дисплеем, экран которого отражает не только процесс ввода, но и результаты обработки информации.

Дисплей — это электронное устройство отображения вводимой и выводимой информации. В персональном компьютере Роботрон 1715 использован монохроматический (одноцветный) дисплей для отображения алфавитно-цифровой информации. На экране дисплея размещаются 24 строки по 80 символов в строке. Изображение воспроизводится зеленым цветом на темном фоне.

Клавиатуру Роботрона 1715 вы освоите достаточно быстро,

немного поработав на ней. В качестве вспомогательного материала, облегчающего процесс ее освоения, используйте рис. 1.5. Рассматриваемые ниже назначения каждой клавиши не зависят от создаваемой программной среды и являются постоянными.

Условно клавиатура разбита на три зоны: буквенно-цифровую, цифровую и функциональную. В основном вы будете работать в буквенно-цифровой зоне, где размещены клавиши русского и латинского алфавитов, различных символов, управляющие клавиши.

Ввод букв русского и латинского алфавитов

Три ряда буквенно-цифровой зоны составляют клавиши русского и латинского алфавитов. На каждой клавише в ее верхней части находится русская буква, а в нижней части — латинская. Расположение букв русского алфавита соответствует стандарту клавиатуры пишущей машинки. Буквы латинского алфавита размещены иначе, чем принято для пишущих машинок. Латинскую букву совмещают на клавишу с той русской буквой, с которой совпадает ее произношение. Например, на одной клавише расположены буквы **П** (русский алфавит) и **P** (латинский алфавит). Буквы русского алфавита образуют верхний регистр, буквы латинского алфавита — нижний регистр. При вводе они отображаются прописными буквами.

Для фиксации верхнего регистра используется черная клавиша без названия, при этом загорается находящийся слева от нее светодиод (индикатор). На рис. 1.5 эта клавиша условно обозначена **ФВР** (фиксация верхнего регистра). При включении компьютера автоматически подключается латинский алфавит, что соответствует погашенному светодиоду клавиши **ФВР**. Нажимая клавишу **ФВР**, вы переходите на верхний регистр с русским алфавитом, о чем свидетельствует загорание светодиода. Для отмены фиксации повторно нажмите эту клавишу. Убе-

дитесь в правильности сказанного, поочередно нажимая клавишу **ФВР** и буквенные клавиши.

Для кратковременного или одноразового перехода на другой регистр воспользуйтесь одной из черных клавишей без названия, для удобства пользования расположенных с левого и правого краев зоны. На рис. 1.5 они условно обозначены **ВПДР** — временное переключение на другой регистр. Ввод с другого регистра возможен только при одновременном нажатии клавиши **ВПДР** и буквы.

Запомните назначение клавиш **ФВР** и **ВПДР**, так как вы будете пользоваться ими постоянно.

Исключение — ввод символов ; — : . , ?

Вводить символы ; — : . , ? вы должны иным, отличным от вышеописанного способом. Независимо от состояния клавиш **ФВР** (светится или не светится светодиод), нажав одну из них, вы введете символ, изображенный в нижней части клавиши. Для ввода символа, расположенного в верхней части клавиши, нажмите одновременно клавишу **ВПДР** и клавишу с этим символом Таким образом, символы ; — : . , ? вводите всегда при нажатой клавише **ВПДР**. Так же можно ввести и цифры верхнего ряда, но на практике для ускорения процесса ввода рекомендуем пользоваться цифровой зоной.

Функции клавиш управления

Функции клавиш управления могут меняться в зависимости от пакета прикладных программ. Здесь вы ознакомитесь с клавишами, действие которых в основном неизменно в операционной среде.

●●● Клавишей ввода ET пользуйтесь при завершении набора строки. После нажатия этой клавиши содержимое строки записывается в оперативную память и курсор перемещается в следующую строку.

●●● CTRL действует только совместно с буквенной клавишей и предназначается для генерации различных

управляющих воздействий. В специальной литературе эту клавишу часто обозначают символом $\wedge$, иногда — кавычками ("). В табл. 1.2 приведены некоторые управляющие функции для операционной среды Роботрона 1715. При изучении пакетов вы часто будете сталкиваться с допол-

Таблица 1.2

Управляющие функции

Клавиша	Функция
CTRL C	Перевод операционной системы в начальное состояние, сброс дисковой системы (дублирует клавишу F14). Кроме того, используется для выхода из обслуживающей команды
CTRL S	Останов вывода на экран дисплея (дублирует клавишу F5)
CTRL M	Дублирует клавишу ET
CTRL U	Отменяет ввод строки (дублирует клавишу CE)
CTRL R	Повторный вывод строки (клавиша F4)
CTRL P	Синхронные печать и вывод на экран
CTRL H	Дублирует клавишу ←
CTRL X	Дублирует клавишу |←

нительными управляющими функциями, воспроизводимыми за счет совместного действия клавиши CTRL и буквенной клавиши. Особенно интенсивно используется клавиша CTRL в пакете прикладных программ по редактированию текста, например в РЕФОРе.

●●● **SI/SO** служит для выбора режима отображения букв латинского и русского алфавитов, чаще используется при работе с русским текстом Об установке этого режима вам сигнализирует светящийся светодиод, установ-

ленный рядом с этой клавишей. Для ввода строчных букв русского алфавита нажмите клавишу **SI/SO** и проверьте, не нажата ли клавиша **ФВР**; об отсутствии нажатия сигнализирует ее погашенный светодиод. Для ввода прописных букв нажмите клавишу **ФВР**, при этом засветится ее светодиод.

●●● Клавиша |← стирает набранную строку. Клавиша ← стирает последний набранный символ и передвигает курсор на предыдущую позицию.

●●● Клавиша табуляции —→ передвигает курсор на восемь позиций вправо.

●●● Клавиша (R) — автоповтор символа, клавиша которого нажата в данный момент. Действует она только при одновременном нажатии клавиши (R) и любой буквенно-цифровой клавиши.

●●● Клавиша **DEL** удаляет символ слева от курсора с одновременным отображением его на экране. Функция этой клавиши может быть изменена при использовании в среде прикладного обеспечения.

●●● Остальные управляющие клавиши буквенно-цифровой зоны действуют только в среде, создаваемой конкретным пакетом прикладных программ.

Клавиши цифровой зоны

Числа набираются в цифровой зоне. Здесь же расположены клавиши с запятой (,) для отделения целой части от дробной и клавиша со знаком минус (—).

●●● Клавиша **S** дублирует действие клавиши **ET**.

●●● Клавиша **CE** отменяет ввод. При ее нажатии в набираемой строке на экране появляется символ # и курсор переходит на следующую строку. Эта функция дублируется с помощью **CTRL U**.

Функциональные клавиши

Большинство функциональных клавиш используется конкретной прикладной программой. Однако есть несколь-

ко клавиш, которые предназначены для управления работой в операционной системе.

●●● **F4** повторно выводит строку ввода на экран. Дублирует действие этой клавиши **CTRL R**. Рекомендуем использовать эту клавишу, когда при наборе строк вы несколько раз исправляли содержимое и на экране зафиксировались все ваши действия. Нажав клавишу **F4**, вы получите строку ввода без исправлений в конечном виде.

●●● **F5** останавливает вывод информации на экран. Дублирует действие этой клавиши **CTRL S**.

●●● **F10** — прогон листа. При работе печатающего устройства обеспечивает переход к началу нового листа.

●●● **F14** переводит операционную систему в начальное состояние, осуществляет сброс дисковой системы. Если вы хотите произвести замену дисков, то после замены обязательно нажмите либо эту клавишу, либо дублирующую ее действие **CTRL C**. В противном случае вы не сможете записать информацию на новый диск.

1.4. ВОЗМОЖНОСТИ ПЕЧАТАЮЩЕГО УСТРОЙСТВА

Вы получаете в руки универсальное средство печати

Печатающее устройство (принтер) обладает широкими возможностями по организации печати текстовой информации или графических изображений. Работая на персональном компьютере Роботрон 1715, вы получаете мощное средство печати с многочисленными вариантами шрифтов и способами оформления текстов. Можно, например, выделять в тексте слова, печатая их вразрядку, жирным или сжатым шрифтом, а также используя подчеркивание. Можно регулировать расстояние между строками и размещать текст в нижней или верхней части одной строки. Различные комбинации шрифтов и умелое управление режимами печати позволяют печатать тексты и рисунки, удовлетворяющие любому взыскательному вкусу.

В этом параграфе вы ознакомитесь с основными способами организации текстовой печати. Для более глубокого изучения средств управления печатью текста и графических изображений вам необходимо обратиться к техническому описанию печатающего устройства [7], поставляемому вместе с аппаратной частью Роботрона 1715.

Техническая характеристика печатающего устройства

В качестве печатающего устройства используется матричное знакосинтезирующее устройство. Головка печати состоит из девяти иголок, при разном сочетании которых воспроизводится символ в требуемом изображении. При наличии хорошей красящей ленты организуется высококачественная печать символов, которые представляют собой различное сочетание микроскопических точек. Скорость печати порядка 200 знаков в секунду. Максимальное количество знаков в строке зависит от выбранного шрифта и типа печатающего устройства. Например, при стандартной печати для принтера К 6312-М максимальное количество символов в строке — 136, а при печати сжатым шрифтом — 233. Расстояние между строками можно регулировать от 2,5 мм до любого значения. Максимальная ширина листа бумаги — 420 мм.

Обратите внимание! В технической литературе по компьютерной технике в качестве меры длины часто используется дюйм (1 дюйм≈≈25,4 мм).

Для большей наглядности в дальнейшем изложении будем пользоваться привычной для нас мерой, выраженной в миллиметрах. Например, механическим способом (переключателями) интервал между строками устанавливается 1/8 или 1/6 дюйма, что соответствует 3,18 и 4,23 мм.

При сравнении скорости работы печатающего устройства с быстродействием центральной части ЭВМ вы заметите существенную разницу. В целях устранения этого несоответствия используется промежуточное устройство для на-

копления информации, подлежащей печати, — буфер печати как одна из разновидностей управляющей памяти. Буфер печати принимает информацию из центральной части компьютера, накапливает ее и по мере готовности печатающего устройства выдает на печать.

Назначение клавиш и индикации

●●● Состояние печатающего устройства характеризуется двумя режимами: рабочим (ON LINE), когда организована печать информации, и нерабочим (OFF LINE). На передней панели печатающего устройства (рис. 1.6)

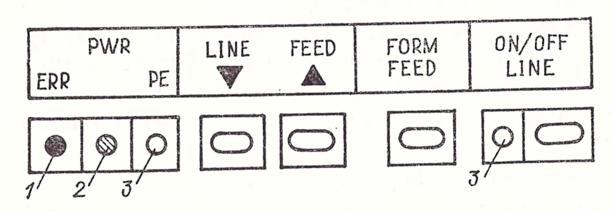


Рис. 1.6. Клавиши управления на передней панели печатающего устройства Роботрона 1715
Подсветка индикатора: *1* — красная; *2* — зеленая; *3* — желтая

находятся ряд клавиш и светодиоды индикации для управления работой и сигнализации о его состоянии. Светодиоды имеют различную подсветку — красную, желтую, зеленую в зависимости от состояния принтера:

красная (ERR — ERROR) — техническая неисправность;
зеленая (PWR — POWER) — готовность к работе;
желтая (PE — PAPER END) — конец бумаги.

Желтое свечение рядом с клавишей **ON/OFF** LINE — принтер в рабочем режиме ON LINE. При погашенном светодиоде — принтер в режиме OFF LINE.

●●● Клавишей **ON/OFF LINE** вы будете пользоваться в том случае, если переключатель 5-1 находится в положении OFF. Тогда для подключения печатающего устрой-

ства в работу (режим ON LINE) вам необходимо нажать эту клавишу, проверив предварительно, закрыта ли крышка принтера. При изменении состояния переключателя 5-1 будет осуществлен автоматический переход принтера в режим ON LINE, и соответствующая клавиша не играет никакой роли (см. табл. 1.3 и рис. 1.7).

●●● Клавиша **FORM FEED** предназначена для прогона до конца листа стандартного формата. Формат — это стандартные длина и ширина листа печати. Например, при печати текста каретка принтера остановилась посередине формата. Для перехода к следующему листу печати нажмите клавишу **FORM FEED**.

●●● Клавиши **LINE FEED.** Нажимая клавишу ▲, вы продвигаете лист бумаги вперед. Нажав клавишу ▼, можно продвинуть лист бумаги назад. Те же операции вы можете проделать, пользуясь находящейся на валике ручкой перевода.

Управление печатью переключателями

Открыв крышку, вы увидите 14 переключателей с обозначениями 5-18, а посередине панели изображены две стрелки: левая стрелка помечена буквами OFF, правая — буквами ON (рис. 1.7). Каждый переключатель имеет две

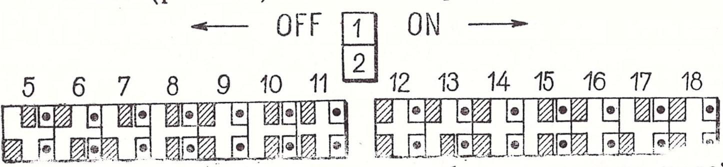


Рис. 1.7. Переключатели режимов работы печатающего устройства

клавиши: верхняя обозначена цифрой 1, нижняя — цифрой 2. Каждая клавиша находится в одном из двух положений — левом (OFF) или правом (ON). Вы можете, например, с помощью переключателя 10 выбрать шрифт, а переключателем 12 задать расстояние между строками. Табл. 1.3 ознакомит вас с функциями этих переключателей.

Таблица 1.3

Функции переключателей

Переключатель	Положение переключателя			Функция
	Номер	OFF	ON	
5	1	+		Контакт крышки действует
	1		+	Контакт крышки не действует
	2			Не используется
6	1	+		Обеспечен концевой контакт бумаги
	1		+	Не обеспечен концевой контакт бумаги
	2	+		Обеспечен сигнал зуммера
	2		+	Не обеспечен сигнал зуммера
7	1	+		Отсутствует автоматическое переключение строк. Требуется программное управление
	1		+	Автоматическое переключение строк реализовано, если переключатель 7-2 находится в положении OFF
	2	+		Автоматический возврат каретки к левому краю, если переключатель 7-1 находится в положении ON
	2		+	Отсутствует автоматический возврат каретки. Требуется программное управление
8	1	+		Не используется
	1		+	Строчные буквы русского алфавита
	2	+		Код управления ESC/P
	2		+	Код управления ИСО
9	1	+		Строчные буквы русского алфавита
	1		+	Прописные буквы латинского и русского алфавитов

Переключатель	Положение переключателя			Функция
	Номер	OFF	ON	
9	2			Не используется
				Виды шрифтов:
10	1	+		стандартный («пика»)
	2	+		
	1		+	«элита»
	2	+		
	1		+	сжатый
	2		+	
	1	+		жирный
	2		+	
11	1	+		Печать числа нуль в виде 0
	1		+	Печать числа нуль в виде ∅
	2	+		Интервал между строками 4,23 мм
	2		+	Интервал между строками 3,18 мм
				Длина формата листа:
12	1	+		304,8 мм (12 дюймов)
	2	+		
	1		+	279,4 мм (11 дюймов)
	2	+		
	1	+		152,4 мм (6 дюймов)
	2		+	
	1		+	139,7 мм (5,5 дюймов)
	2		+	
13	1	+		Подача бумаги на 25,4 мм за фальц. При непрерывной печати этот прогон бумаги разграничивает соседние листы печати
	1		+	Не действует
	2			Не используется

Переключатель	Положение переключателя			Функция
	Номер	OFF	ON	
14 . . . 18				Предназначены для организации интерфейса между принтером и системным блоком. Настройка производится на конкретную модель [7]

В ней установка переключателя в требуемое положение OFF или ON обозначается знаком +.

Предостережение! Пользуйтесь переключателями только в случае крайней необходимости. Рекомендуем применять в качестве основного способа программное управление режимом печати. Частое механическое воздействие может привести к неисправности как переключателей, так и печатающего устройства.

Управление печатью программным способом

Управление печатью удобно производить из программы, используя соответствующие средства алгоритмического языка. Возможности программного способа гораздо шире, и он более мобилен, нежели управление с помощью переключателей. Кроме того, при таком способе отсутствует механическое воздействие на принтер. Для освоения всех способов печати рекомендуем воспользоваться инструкцией [7]. Здесь же будет показан основной способ задания шрифтов, ознакомившись с которым, вы легко освоите и другие способы.

Воспользуйтесь интерпретатором алгоритмического языка Бейсик, загрузив его в оперативную память, и рекомендациями параграфа 1.5.

◆ Поставьте диск, на котором записана операционная система и Бейсик, в дисковод А.

◆ Начнется автоматическая загрузка операционной системы. После ее окончания на экране появится подсказка

A>

◆ Наберите слово BASIC и нажмите клавишу **ET**

A>BASIC

◆ После загрузки транслятора Бейсик снова появится подсказка A>.

Теперь, находясь в среде языка Бейсик, для программного управления печатью вы будете пользоваться оператором LPRINT, после которого должен следовать код выбранного способа печати.

●●● Предлагаемый вашему вниманию один из способов обеспечивает печать текста выбранным шрифтом. Отмена выбранного варианта печати производится заданием нового кода шрифта. Совокупность кодов

CHR¤(27); "!"; CHRS(m)

реализует один из возможных видов шрифтов: стандартный, часто называемый «пика», «элита», сжатый, жирный с использованием стандартной или двойной печати, а также вразрядку. Вид шрифта определяется кодом из табл. 1.4, который вы должны подставить вместо указанной в совокупности кодов буквы m. На рис. 1.8 показаны все возможные виды шрифтов.

Таблица 1.4

Коды шрифтов

Шрифт	Печать			
	стандартная	двойная	вразрядку	двойная вразрядку
Стандартный	0	16	32	48
«Элита»	1	17	33	49
Сжатый	4	20	36	52
Жирный	8	24	40	56

●●● Интервал между строками устанавливается также оператором LPRINT, за которым следует совокупность кодов, представленная в табл. 1.5.

Например, вы хотите напечатать слово «ПРОГРАММА» сначала жирным шрифтом, а затем шрифтом «элита» вразрядку. Интервал между строками должен быть обеспечен приблизительно 20 мм, для чего, исходя из

Таблица 1.5

Коды интервалов между строками

Интервал, мм (дюйм)	2,5 (7/72)	3,18 (1/8)	4,23 (1/6)	n·0,12 (n/216)	n·0,35 (n/72)
Код	CHR¤(27); "1"	CHR¤(27); "0"	CHR¤(27); "2"	CHR¤(27); "3"; CHR¤(n)	CHR¤(27); "A"; CHR¤(n)

СТАНДАРТНЫЙ ШРИФТ
-ДВОЙНОЙ
-ВРАЗРЯДКУ
-ДВОЙНОЙ ВРАЗРЯДКУ

ШРИФТ "ЭЛИТА"
-ДВОЙНОЙ
-ВРАЗРЯДКУ
-ДВОЙНОЙ ВРАЗРЯДКУ

СЖАТЫЙ ШРИФТ
-ДВОЙНОЙ
-ВРАЗРЯДКУ
-ДВОЙНОЙ ВРАЗРЯДКУ

ЖИРНЫЙ ШРИФТ
-ДВОЙНОЙ
-ВРАЗРЯДКУ
-ДВОЙНОЙ ВРАЗРЯДКУ

Рис. 1.8. Варианты шрифтов

табл. 1.5, следует определить n=20/0,35≈5,7≈6 мм. Тогда программа управления печатью примет вид:

10 LPRINT CHR ¤ (27); "!"; CHRS (8) 20 LPRINT "ПРОГРАММА"	Печатать жирным шрифтом
30 LPRINT CHR ¤ (27); "A"; CHR ¤ (6)	Интервал между строками
40 LPRINT CHR ¤ (27); "!"; CHRS (33) 50 LPRINT "ПРОГРАММА"	Печатать шрифтом «элита» вразрядку

1.5. КАК ПРИСТУПИТЬ К РАБОТЕ НА РОБОТРОНЕ 1715?

Включите компьютер

Включите персональный компьютер Роботрон 1715 в сеть. Затем переведите переключатель, находящийся на передней панели системного блока, в положение **ON** (см. рис. 1.2).

Подключите печатающее устройство

При необходимости печати подключите печатающее устройство к работе с помощью клавиши включения, находящейся на его задней панели справа. Печатающее устройство автоматически подключается в нормальный режим работы ON LINE, и загорается желтый светодиод. Однако возможна ситуация, когда автоматического подключения принтера не произойдет. Это может быть либо по причине плохо закрытой крышки или отсутствия бумаги. Для того чтобы убедиться в правильности предположения, откройте крышку и посмотрите, в каком положении находятся переключатели 5-1, 6-1. Если в положении OFF,

то устраните причину, закрыв крышку и заправив бумагу, а затем нажмите клавишу **ON/OFF LINE**. Если в положении ON, то причина будет иной, и скорей всего — это неисправность печатающего устройства.

При отсутствии потребности в печати принтер не включайте!

Загрузите операционную систему

Следующим шагом является загрузка операционной системы. Вставьте диск с операционной системой в дисковод A и закройте защелку. Начнется автоматическая загрузка операционной системы. После ее окончания на экране появится название версии операционной системы и подсказка:

ROBOTRON 1715
SCP VERS. XX—YY. YY. YY.—ZZ KB
A> _

Здесь условно обозначены: XX — номер версии; YY.YY.YY — дата составления; ZZ — размер свободной зоны оперативной памяти. Подсказка **A>** означает, что левый дисковод с именем A определен актуальным. Символ _ (курсор) указывает позицию вводимого символа.

Замечание! Операционная система всегда загружается с левого дисковода A.

Загрузите пакет прикладных программ

Загрузить в оперативную память пакет прикладных программ можно несколькими способами в зависимости от того, на каком диске находится операционная система.
●●● Предположим, что операционная система и пакет прикладных программ находятся на разных дисках. Тогда откройте защелку дисковода A, выньте диск с операционной системой, вставьте туда же диск с пакетом прикладных программ и закройте защелку дисковода.

Далее вызовите каталог диска, набрав команду DIR и нажав клавишу **ET**:

A>DIR

Для загрузки пакета прикладных программ вам надо найти в каталоге имя ведущего файла. Затем наберите это имя и нажмите клавишу ввода **ET**:

A>имя файла

Таким образом, вы вошли в среду пакета.

●●● В том случае, когда операционная система и пакет прикладных программ находятся на одном диске, вставьте этот диск в дисковод и проделайте все указанные выше действия, начиная с ввода команды вызова каталога.

●●● Возможен также вариант, когда вы хотите сохранить системный диск в дисководе A. Тогда вставьте диск с пакетом прикладных программ в дисковод B и закройте защелку. Далее сделайте дисковод актуальным (текущим), вводя строку

A>B:

На экране появится подсказка

B>_

Далее вы вводите либо имя пакета, либо, если вы забыли его, воспользуйтесь командой DIR для просмотра каталога диска, а затем введите имя.

1.6. ПРОВЕРЯЕМ СЕБЯ. ЧТО ПРЕДСТАВЛЯЮТ СОБОЙ АППАРАТНЫЕ СРЕДСТВА?

Упражнения

1.1. Из каких функциональных блоков состоит структурная схема Роботрона 1715?

1.2. В чем особенности аппаратной реализации структурной схемы Роботрона 1715?

1.3. Состав системного блока.

1.4. Для чего нужен гибкий диск?

1.5. Как вы понимаете назначение дисковода?

1.6. Какие дисководы могут входить в комплект поставки Роботрона 1715?

1.7. Как вводить буквы русского алфавита?

1.8. Как вводить буквы латинского алфавита?

1.9. Назначение клавиши **CTRL**.

1.10. Как вводятся символы . , : — ; ?

1.11. Что свидетельствует о готовности печатающего устройства к работе?

1.12. Как можно управлять работой печатающего устройства?

1.13. В чем состоят возможности печатающего устройства?

1.14. Как выглядит программная строка задания печати определенным шрифтом?

1.15. Какова последовательность ваших действий для работы с программой, хранящейся на диске?

Ответы к упражнениям

1.1. Структурная схема персонального компьютера Роботрон 1715 отражает традиционный подход к архитектуре микроЭВМ, основанный на использовании интерфейса «общая шина». В структуре персонального компьютера можно выделить центральную часть, состоящую из процессора и внутренней памяти, и периферийную часть, состоящую из устройств связи центральной части с внешней памятью и пользователем. Все устройства соединяются системной шиной.

1.2. Аппаратная реализация структурной схемы не имеет четко выраженной центральной и периферийной частей, как в классе больших ЭВМ. Конструктивно персональный компьютер Роботрон 1715 выполнен в виде четырех автономных блоков: системного блока, клавиатуры, дисплея, печатающего устройства.

1.3. В состав системного блока входят: системная плата, на которой реализована центральная часть компьютера, адаптерная плата, где установлены контроллеры для управления периферийной частью; дисководы; источник питания и вентилятор.

1.4. Гибкий диск служит внешней памятью персонального компьютера для долговременного хранения программ и данных.

1.5. Назначение дисковода — организация записи информации на гибкий диск и чтение информации с диска.

1.6. Роботрон 1715 может быть укомплектован дисководами одного из перечисленных ниже типов:

MFS 1.2 обеспечивает одинарную плотность на одной стороне диска с 40 дорожками;

MFS 1.4 обеспечивает двойную плотность на одной стороне диска с 80 дорожками;

MFS 1.6 обеспечивает двойную плотность на обе стороны диска по 80 дорожек.

1.7. Предварительно нажмите клавишу фиксации верхнего регистра (**ФВР** на рис. 1.5), чтобы засветился индикатор, а затем нажмите буквенную клавишу. Если вы постоянно работаете с нижним регистром, вводя латинские буквы, то для единовременного ввода русской буквы нажмите одновременно клавишу **ВПДР** (см. рис. 1.5) и буквенную клавишу.

1.8. Необходимо, чтобы индикатор клавиши фиксации верхнего регистра был погашен. Если вы работаете на верхнем регистре, то для единовременного ввода латинской буквы нажмите одновременно клавишу **ВПДР** и соответствующую буквенную клавишу.

1.9. Клавиша **CTRL** действует совместно с буквенной клавишей для генерации управляющего воздействия.

1.10. Необходимо одновременно нажать клавишу **ВПДР** и клавишу с нужным символом.

1.11. Светятся индикаторы: зеленый (PWR) и желтый под клавишей **ON/OFF LINE**.

1.12. С помощью переключателей 5-18, находящихся под крышкой печатающего устройства, или посредством кодов программным способом. Предпочтителен программный способ.

1.13. Возможности печатающего устройства:

- печать разными шрифтами — стандартным («пика»), «элита», сжатым, жирным — обычным способом или вразрядку, а для большей четкости применяется двойная печать;
- регулируется расстояние между строками;
- при печати можно задать разный формат листа.

1.14. Программная строка на алгоритмическом языке Бейсик состоит из оператора LPRINT с кодами в соответствии с табл. 1.4. Эта строка должна предшествовать выводимой информации. Например, для печати слова «КОНТРОЛЬ» жирным шрифтом вразрядку программная строка имеет вид:

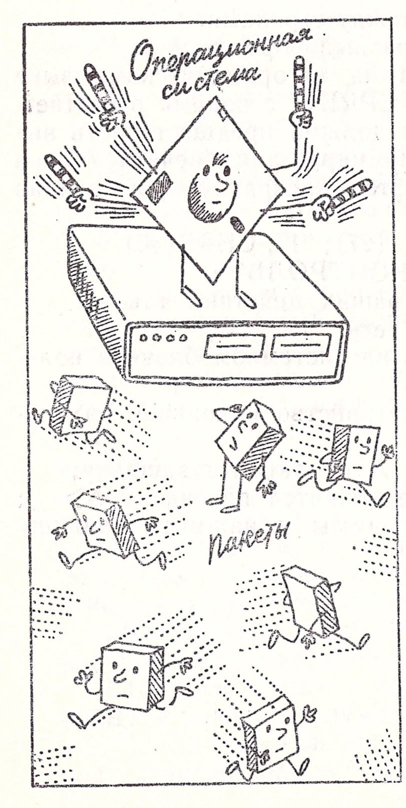
```
10 LPRINT CHR ¤ (27); "!"; CHRS(40)
20 LPRINT "КОНТРОЛЬ"
```

1.15. Последовательность ваших действий такова:

- включите Роботрон 1715 в сеть;
- переведите переключатель на системном блоке в положение ON;
- подключите печатающее устройство клавишей, находящейся на его задней панели;
- вставьте диск в дисковод A и закройте защелку;
- ждите, пока на экране не появится подсказка A>__;
- наберите имя вашей программы и нажмите клавишу ввода ET.



ИЗУЧАЕМ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РОБОТРОНА 1715



2.1. ВОЗМОЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Что следует понимать под программным обеспечением?

Ознакомившись с аппаратной частью компьютера, вы уже достаточно ясно представляете функции составляющих его блоков. Но как же организовать взаимодействие блоков, как использовать их возможности для достижения поставленной цели, какие нужны средства для организации процесса обработки информации на компьютере? Ответы на эти вопросы можно получить, четко осознав, что основополагающим фактором могущества компьютерной техники являются

программы, разработанные и заложенные в память компьютера человеком, и от того, насколько они совершенны, зависят эффективность работы аппаратной части и успешное решение поставленных задач.

При решении вопроса о приобретении той или иной модели персонального компьютера опытный специалист прежде всего поинтересуется назначением и структурой стандартных программных средств, входящих в комплект поставки. Кроме того, он выяснит, какие имеются дополнительные программные средства и насколько они развиты, существуют ли реальные возможности их приобретения. Безусловно, предпочтение будет отдано модели с бóльшим числом универсальных программ обработки информации различного назначения.

Совокупность программ общего назначения, организующую обработку информации на компьютере, называют его программным обеспечением. Некоторые программы входят в состав компьютера и поставляются на дисках вместе с аппаратурой. Однако же, как правило, это ничтожная часть всего многообразия программных средств. В основном программное обеспечение вы должны покупать у распространяющих его организаций. В нашей стране центральной организацией, куда в основном обращаются владельцы компьютеров, является научно-производственное объединение «Центрпрограммсистем» в городе Калинине. Кроме того, организуются подобные центры регионального назначения, например в Таллине, Ленинграде.

Программное обеспечение, независимо от конкретной модели компьютера, подразделяется на три класса: системное, система программирования, прикладное. Однако это не означает, что, имея, например, прикладное программное обеспечение для персонального компьютера, вы можете использовать его в мини-ЭВМ или больших ЭВМ. Программное обеспечение может быть общим только для компьютеров с единой концепцией построения. Так, про-

граммное обеспечение больших ЭВМ с успехом применяется в различных моделях ЕС, программное обеспечение персональных компьютеров является общим, если в их структуре используется один и тот же тип микропроцессора, и неприемлемо для моделей, созданных на базе

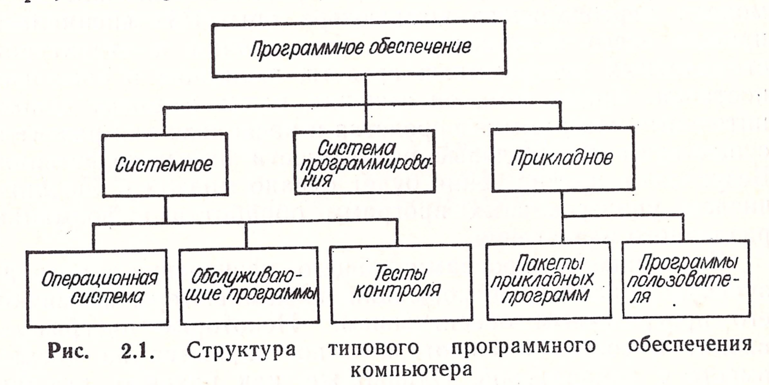


Рис. 2.1. Структура типового программного обеспечения компьютера

разных типов микропроцессоров. Однако это ограничение может быть снято, если дополнительно используются специальные программные и технические средства. Программное обеспечение Роботрона 1715 представлено на рис. 2.1 в соответствии с общей классификацией без отражения специфики конкретной модели.

Системное программное обеспечение

Системное программное обеспечение любого компьютера управляет обработкой информации и контролирует работу аппаратной части. В состав системного программного обеспечения Роботрона 1715 входят операционная система SCP, обслуживающие команды, тесты контроля за работой устройств (см. рис. 2.1).

●●● Операционная система SCP и обслуживающие команды организуют обработку информации на компьютере и управляют его аппаратной частью. Это неотъемлемый

элемент общей системы управления, без которой невозможно ни одно действие компьютера. Операционную систему можно считать практически неизменяемой частью с ограниченным набором команд для реализации основных функций управления. Однако ее возможности вы можете существенно расширить за счет использования обслуживающих (сервисных, транзитных) команд, хранящихся независимо от операционной системы в виде файлов на дисках.

●●● Тесты контроля за работой устройств представляют собой специальные программы, с помощью которых вы проверяете работоспособность отдельных блоков и при неисправности выявляете причину ее возникновения.

Система программирования

Система программирования позволяет разрабатывать программы на символических языках, а не в машинных кодах, где инструкции компьютеру (команды) о проведении им тех или иных действий выражены двоичными кодами — различными сочетаниями двух цифр: 1 и 0.

Напоминаем! Программа — это совокупность инструкций, написанных человеком, для выполнения компьютером определенных действий.

Что представляют собой символические языки и каково их назначение? Символические языки должны обеспечить понимание компьютером задач пользователя. Концепция их построения и использования схожа с разговорным языком. Вспомните, как обучаются дети читать и писать. Сначала их знакомят с алфавитом языка, а затем, постепенно овладев правилами образования (синтаксисом) слов, предложений, различных конструкций, они приобретают способность свободно читать, правильно выражать свои мысли, грамотно писать. Итак, важнейшим средством общения между людьми является естественный язык. Общение же человека и компьютера призвана обеспечить среда символического языка.

Символические языки также в основе своей имеют алфавит, но в них предусмотрены не только буквы, но и специальные символы. В основу образования различных конструкций символического языка также заложены специальные правила — синтаксис. Символические языки подразделяются на алгоритмические и машинно-ориентированные.

●●● По своей структуре алгоритмические языки не имеют ориентации на конкретную модель компьютера, и поэтому программы, написанные на каком-то определенном алгоритмическом языке, с небольшими доработками, учитывающими специфику устройств ввода-вывода, вы можете использовать на разных компьютерах. Алгоритмические языки образуют многочисленное семейство, насчитывающее порядка тысячи представителей. Наибольшее распространение получили языки Бейсик, Паскаль, ПЛ/1, Ада, Фортран.

●●● Машинно-ориентированные языки существенно сложнее, так как учитывают структуру конкретной модели компьютера. В связи с этим и программы, написанные на этих языках, рассчитаны только на те модели, где можно использовать идентичные наборы команд. К этой группе языков относятся Ассемблер, Макроассемблер, Автокод и др.

Однако, поскольку компьютер способен понимать только машинный язык, необходимо хранить в его памяти комплекс программ, которые обеспечат перевод всех конструкций алгоритмического языка в машинный язык конкретной модели компьютера с учетом специфики его структуры. Такой комплекс программ называют транслятором. Организация работы транслятора может идти двумя путями — интерпретации или компиляции программ. Соответственно и транслятор будет называться либо интерпретатором, либо компилятором. Различие их состоит в разной процедуре обработки конструкций программы.

●●● Интерпретатор производит анализ, перевод и выполнение операторов, составляющих программу, один за другим. После интерпретации очередного оператора происходит переход к обработке следующего. Таким образом, например, выполнение в цикле какого-либо оператора (или группы операторов) потребует его анализа и перевода столько раз, сколько выполняется цикл. Как вы понимаете, это приводит к замедлению процесса выполнения программы, что особенно заметно при циклической обработке данных.

Безусловно, такой путь трансляции не очень эффективен. Однако интерпретатор обладает очень важными качествами, благодаря которым он получил широкое распространение. Интерпретатор особенно полезен при отладке программ. Работа с интерпретатором языка производится в диалоговом режиме путем выдачи им сообщений на экран дисплея о допущенных вами ошибках при составлении программы. Многие интерпретаторы позволяют обнаруживать ошибки непосредственно во время ввода исходного текста программы. Причем продолжение интерпретации программ возможно только после того, как вы внесете исправления.

Практически любой интерпретатор позволит вам выполнить программу начиная с любого оператора и просмотреть содержимое переменных, полученное в результате выполнения программы. Основное удобство при работе с интерпретатором состоит в том, что вы получаете возможность в диалоговом режиме внести исправления в программу и повторно ее выполнить без каких-либо задержек.

●●● Компилятор, в отличие от интерпретатора, производит перевод целиком всей исходной программы, а затем только ее проверяет и выполняет. Если введенная вами программа не содержит ошибок, то дальше ничто не мешает ее выполнить. В противном случае вам следует исправить обнаруженные ошибки и повторить компиляцию.

Что дает компилятор? Прежде всего он обеспечивает более быстрое, по сравнению с интерпретатором, выполнение программы за счет того, что анализ и перевод операторов, составляющих программу, выполняется один раз. Другое преимущество компилятора заключается в том, что, в отличие от интерпретатора, он не нужен в процессе выполнения программы.

●●●Вывод может быть сделан следующий — в процессе отладки программы целесообразно пользоваться интерпретатором. Отлаженную же программу лучше всего обработать компилятором для обеспечения большей скорости ее выполнения.

В компьютере Роботрон 1715 используются компиляторы с языков: Бейсик, Паскаль, Си, Ада, Фортран, Ассемблер. Кроме того, широкое применение как для создания программ, так и в целях обучения программированию нашел интерпретатор языка Бейсик.

Вы, вероятно, зададите вопрос: каким же языком программирования вам лучше овладеть? Однозначного ответа дать невозможно, так как это зависит от цели поставленной задачи, возможностей конкретного языка, навыков и умения программировать. Если вы никогда раньше не имели дела с языком программирования, то целесообразно изучить Бейсик и работать с интерпретатором. При желании стать профессиональным программистом, наверно, следует изучить язык Паскаль. На сегодняшний день это наиболее хорошо структурированный язык с большими возможностями. Языки Ассемблер, Си, Ада, Фортран понадобятся, безусловно, только профессиональным программистам, постоянно работающим над созданием программного обеспечения.

Прикладное программное обеспечение

Прикладное программное обеспечение персональных компьютеров составляют пакеты прикладных программ и уникальные программы пользователя (см. рис. 2.1).

Пакет прикладных программ (ППП) — это комплекс программ для решения определенного класса задач. Существуют три разновидности пакетов:

- методоориентированные для реализации метода решения задачи без привязки к конкретной области, например: для обработки статистических данных или решения задач линейного программирования;
- проблемноориентированные для реализации проблем определенной области. Например, для организации бухгалтерского дела.
- общего назначения; например, для создания текстов используется текстовый редактор РЕФОР, для представления данных в табличной форме и ведения в ней расчетов — пакет прикладных программ ВАРИТАБ.

Важной характеристикой качества работы пакета прикладных программ служит возможность создания им для вас «дружественной» обстановки при работе на компьютере, т. е. такой обстановки, при которой вы чувствуете себя комфортно и над вами не довлеет чувство несостоятельности перед незнакомой техникой. Это достигается за счет организации диалога между вами и компьютером на вашем родном языке, причем темп диалога будете задавать вы. Для работы в среде подобного пакета вам предварительно надо изучить входной язык, т. е. команды, реализующие функции пакета, и правила их использования. Входной язык пакета в отличие от алгоритмических языков, как правило, очень прост, и достаточно, например, внимательно прочитать эту книгу, чтобы овладеть основными навыками работы с подобными пакетами.

Для Роботрона 1715, построенного на базе 8-разрядного микропроцессора, можно использовать все программные средства, ориентированные на работу с таким процессором. За рубежом имеется солидный перечень пакетов прикладных программ для этого класса компьютеров, хотя сейчас в основном выпускаются и ведутся перспективные разработки для 16- и 32-разрядных микропроцессов.

Конечно, для работы с пакетом, приобретенным за рубежом, необходимо знание языка страны-изготовителя, что, безусловно, является препятствием на пути его внедрения. Некоторые из наиболее распространенных пакетов в настоящее время переведены на русский язык и называют их адаптированными пакетами. Эти пакеты прикладных программ находятся на сопровождении в научно-производственном объединении «Центрпрограммсистем» и других региональных центрах, а в настоящей книге показано, как с ними работать. Несмотря на то что перечень рассматриваемых пакетов весьма ограничен, овладев навыками работы с ними, вы будете весьма успешно решать многие проблемы в вашей области. Ведется большая работа по созданию более совершенных прикладных программных средств, и, возможно, в ближайшие годы вы сможете их приобрести, обратившись в соответствующие центры по их распространению.

2.2. ОПЕРАЦИОННАЯ СИСТЕМА — ГЛАВНЫЙ УПРАВЛЯЮЩИЙ ОРГАН КОМПЬЮТЕРА

Операционная система SCP является аналогом широко распространенной в мире операционной системы CP/M (Control Program for Microcomputers), созданной в 1974 г. Гарри Килдаллом (США) для 8-разрядного процессора. Несмотря на то что с тех пор эта система претерпела некоторые изменения, основная концепция ее построения осталась неизменной и заложена в основу более развитых операционных систем для 16- и 32-разрядных микропроцессоров. Система рассчитана на работу одного пользователя и обладает очень надежной файловой системой.

2.2.1. ПРЕДСТАВЛЕНИЕ О ФАЙЛОВОЙ СИСТЕМЕ

Для чего нужен файл и как образуется его имя?

Файл — это совокупность логически связанных записей, занимающая определенное место во внешней памяти. Записями могут служить данные или фрагменты программ. А значит, файлом может быть представлена любая программа или совокупность данных, обрабатываемая определенной процедурой, программой или командой.

Каждый файл имеет уникальное полное имя, состоящее из имени файла и типа, разделенных точкой. Имя файла образуется максимум из восьми символов, исключая символы . * ?. Преимущественно используются латинские буквы или цифры. Тип файла служит для характеристики хранящихся в нем записей и образован максимум из трех символов, исключая символы . * ?. Причем тип файла обязательно должен быть согласован с использующими его программами. Например, существует договоренность относительно назначения некоторых файлов, которая выражается соответствующим стандартным типом:

COM — обслуживающая команда операционной системы в машинном коде;

MAC — программа на языке ассемблера;

BAS — программа на языке Бейсик;

HLP — текст, позволяющий пользователю при работе с пакетом прикладных программ получить разъяснение по командам входного языка.

Посмотрите, как записывается полное имя:

GRUPPA. MAC; PIP. COM; A.BAS; KOT.A

Как быть в том случае, если файл находится на диске в неактуальном дисководе, а вам он необходим для рабо-

ты? Тогда перед полным именем укажите имя другого дисковода, отделив его двоеточием. Например,

B: REFOR. HLP

имя дисковода имя файла тип

Обратите внимание! Дисковод будет актуальным (текущим) в том случае, когда автоматически производится процесс чтения и записи информации на диск, установленный в этом дисководе. О том, какой дисковод в текущий момент является актуальным, вы узнаете из подсказки на экране, где указано его имя.

Атрибуты файла

Для указания принципа использования файла введены атрибуты:

- R/W (read/write) — файл разрешено считывать, а также можно записывать или стирать ненужные записи;
- R/O (read/only) — из файла разрешено только считывание;
- DIR (directory) — имя файла выводится на индикацию резидентной командой DIR (см. параграф 2.2.3);
- SYS(SYSTEM) — имя файла не индицируется командой DIR.

В большинстве случаев вы будете использовать файлы с атрибутами R/W, DIR, устанавливаемыми по умолчанию. Изменять атрибуты можно обслуживающей командой STAT (см. параграф 2.2.3).

Обращение к нескольким файлам

При обращении к нескольким файлам вы можете воспользоваться символами * и ?.

Символ * позволяет заменить любую последовательность в полном имени файла при его вызове. Например, задав полное имя в виде * .BAS, вы организуете тем самым обращение ко всем файлам типа BAS. Задав же полное имя в виде WT.*, обращаетесь к любому типу файла с именем WT.

Символ ? может заменить в полном имени любой сим-

вол в указанном месте и служит для активизации нескольких файлов со схожими именами.

Например, при вводе имени файла ABC???.BAS идет обращение ко всем файлам типа BAS, имя которых содержит три—шесть символов, начинающихся с ABC.

Как организована файловая система?

Файловая система — это способ распределения пространства внешней (дисковой) памяти и управление работой этого пространства. Представьте себе, что на диске находится множество файлов, каждый из которых имеет свое уникальное имя. Как же осуществить быстрый доступ к этим файлам, чтобы время ответов на запрос пользователя было малым? Вспомните процедуру поиска книги в библиотеке. Вы обращаетесь к каталогу, находите в нем шифр этой книги. Библиотекарь, пользуясь этим шифром, быстро отыщет в фонде требуемую книгу и принесет ее вам. Смог бы библиотекарь, зная только автора и название книги, найти ее? Квалифицированный библиотекарь, скорей всего, нашел бы ее. Но сколько бы времени ушло на поиск, сколько ненужных действий пришлось бы совершить.

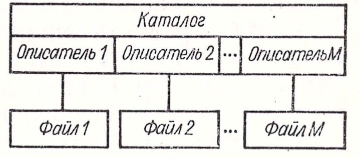


Рис. 2.2. Структура файловой системы

Принцип поиска, основанный на использовании каталога и находящихся в них шифров, лег в основу организации файловой системы. Итак, на диске на третьей дорожке выделено совершенно определенное место (2 Кбайт для дисковода MFS 1.2 и 4 Кбайт для MFS 1.4) под каталог, часто также называемый директорием. В нем хранятся описатели файлов с указанием имен файлов и занятого ими места на диске. Доступ к файлам возможен только через эти описатели, т. е. через каталог (рис. 2.2).

Описатель файла состоит из 32 байт, используемых для

записи полного имени файла, карты распределения диска и вспомогательной информации операционной системы (рис. 2.3). Полное имя файла составляет 11 байт: байты с 1 по 8 занимает имя; байты с 9 по 11 отведены для типа и атрибутов.

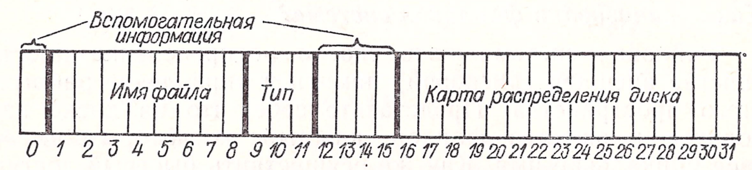


Рис. 2.3. Структура описателя файла

Карту распределения диска составляют номера занятых данным файлом блоков по 2048 байт (16 записей по 128 байт), представленные в шестнадцатеричной системе счисления. Всего отводится для карты 16 байт (с 16 по 31), а каждый номер в виде двух шестнадцатеричных цифр занимает один байт. Таких блоков, например, на диске с односторонней плотностью будет порядка 90. Область размещения файла не обязательно должна быть непрерывной. Один файл может занимать несколько блоков на разных дорожках. В случае увеличения файла производится поиск незанятых блоков и их номера заносятся в карту распределения диска. Вспомогательная информация хранится в байтах с номерами 0,12—15, необходима операционной системе и связана с физическим размещением файла на диске. Таким образом, по имени файла и номерам занятых блоков операционная система найдет место его физического размещения. В том случае, когда требуется организовать файл большого объема, используется несколько описателей. Допустимо иметь для одного файла максимально 16 описателей, причем все они связываются между собой операционной системой.

При стирании файла его блоки поступают в распоряжение других файлов.

2.2.2. СТРУКТУРА И ПРИНЦИП РАБОТЫ ОПЕРАЦИОННОЙ СИСТЕМЫ

Структура операционной системы

Операционная система SCP базируется на трех подсистемах: процессоре консольных команд, базовой дисковой операционной системе и базовой системе ввода-вывода. Организует связь между ними зона связи. При размеще-

Рис. 2.4. Распределение оперативной памяти после загрузки в нее операционной системы

нии на диске операционная система всегда займет дорожки с номерами 0, 1, 2. Обратившись к рис. 2.4, вы увидите распределение зон оперативной памяти после загрузки в нее операционной системы. Все три подсистемы занимают зону, определяемую старшими адресами объемом примерно 16 Кбайт. Однако в процессе работы с прикладными программами объем памяти, отведенный для операционной системы, может быть уменьшен за счет освобождения ее от процессора консольных команд. Освободившаяся часть отдается в ведение прикладного средства. Зона оперативной памяти, определяемая младшими адресами 0000H — 0100H (в шестнадцатеричной системе счисления), — это зона для организации связи трех подсистем.

Основной объем памяти (около 48 Кбайт) отдан в распоряжение прикладных программ и различных обслуживающих программ операционной системы. Эта зона памяти называется областью транзитных программ. На рис. 2.5 показано взаимодействие аппаратного обеспечения ком-

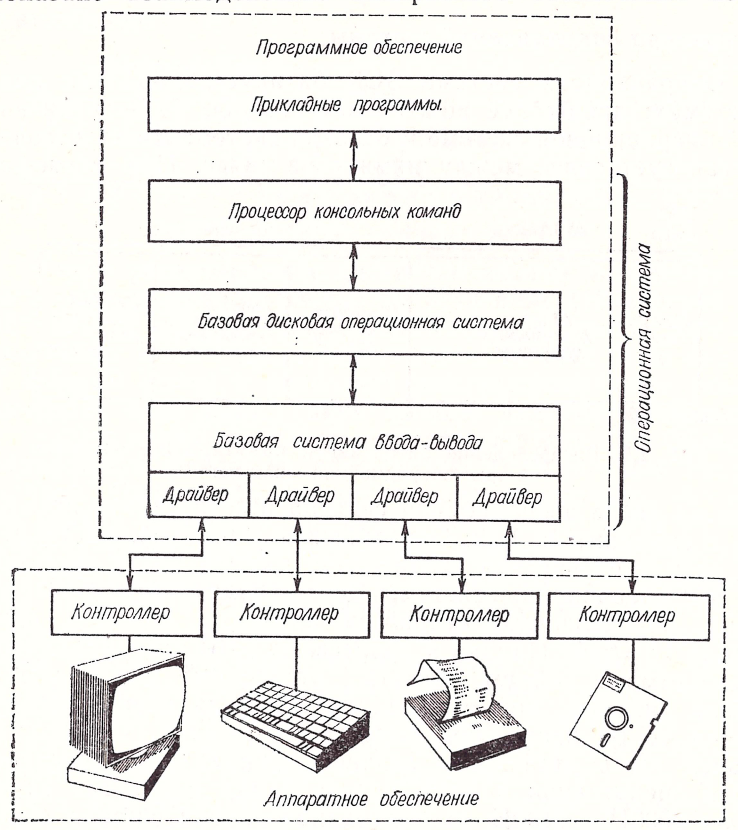


Рис. 2.5. Взаимодействие подсистем операционной системы с аппаратной частью Роботрона 1715

пьютера с его программным обеспечением, состоящим из подсистем операционной системы и прикладных программ.

Процессор консольных команд (ПКК)

Процессор консольных команд служит для организации связи пользователя и компьютера. Он анализирует набираемые вами команды, проверяет их синтаксис и обращается к базовой дисковой операционной системе, например, посылая команды считывания или записи файлов, для выполнения необходимых действий. Наличие на экране подсказки

A>_

свидетельствует о том, что процессор консольных команд ждет от вас указаний в виде команд.

Главная функция процессора консольных команд — загрузка и запуск прикладных программ. Вы вводите название прикладной программы, процессор консольных команд загружает ее и передает управление этой программе. В процессе выполнения прикладной программы при нехватке места в оперативной памяти часть этой подсистемы может быть из нее удалена. По выходе из прикладной программы процессор консольных команд вновь подключается на связь с пользователем.

Базовая дисковая операционная система (БДОС)

Базовая дисковая операционная система является ядром операционной системы, так как ее функции не зависят от конкретной реализации аппаратных средств. В основу работы БДОС заложен принцип организации файловой системы безотносительно к тому, на каком диске хранится или с какого диска считывается файл. В БДОС обеспечена полная независимость операционной системы от аппаратуры. Обращение к ней производится из процессора консольных команд (см. рис. 2.5) для выполнения определенных функций. Основная функция ба-

зовой дисковой операционной системы состоит в организации операций ввода-вывода с диском, т. е. в организации доступа к файловой системе. Например, БДОС организует открытие или закрытие файла, создание или удаление файла, находит имя в каталоге или переименовывает файл, а также выполняет множество других операций по работе с файловой системой.

Базовая система ввода-вывода (БСВВ)

Базовая система ввода-вывода в отличие от двух других подсистем операционной системы предназначена для работы с конкретным составом устройств аппаратной части. Основные функции базовой системы ввода-вывода состоят в следующем:

- прием символов с клавиатуры;
- выдача символов на экран или печать их на принтере;
- организация считывания информации с диска или ее запись на диск;
- определение готовности внешних устройств и установление различных их состояний.

Обеспечить выполнение этих функций возможно только, учитывая специфику каждого устройства. А это требует жесткой зависимости программных средств базовой системы ввода-вывода от конкретного вида внешних устройств. При изменении типа внешнего устройства должна быть изменена и программа связи с ним. Подобные программы, предназначенные для организации обмена данными между внешними устройствами и центральной частью компьютера, называют драйверами. Совокупность драйверов образует базовую систему ввода-вывода. Драйвер включается в работу в тот момент, когда в него из базовой дисковой операционной системы поступает команда (см. рис. 2.5). Тогда он анализирует принятую команду и посылает сигналы в контроллер внешнего устройства, который организует физическую связь внешнего устройства с центральной частью.

Подводя итоги, можно сделать вывод: управление работой внешних устройств осуществляется программно-аппаратным способом, часть функций выполняют программы-драйверы, а часть — схемы-контроллеры.

Например, при работе на клавиатуре контроллер принимает в буферную память код нажатой клавиши, сообщает об этом процессору посылкой сигнала и продолжает прием вводимых символов, заполняя соответствующими кодами свой буфер. По мере освобождения процессор, имея сигнал от контроллера клавиатуры, обращается к нему и принимает все накопленные в буфере коды. Роль драйвера заключается в том, что он управляет обработкой при смене регистра клавиатуры, проверяет, нажимались ли управляющие клавиши, организует процесс записи символов в буфер контроллера и выполняет еще множество функций, которые представляют интерес только для специалистов по компьютерной технике.

Замена модели внешнего устройства влечет за собой замену контроллера и драйвера как двух взаимосвязанных и взаимодополняющих друг друга частей. Обратившись к рис. 2.5, вы увидите, что информация с внешнего устройства попадает в прикладную программу через контроллер и драйвер, подключаемые посредством базовой дисковой операционной системы, а при необходимости — и процессором консольных команд.

Как работает операционная система?

После загрузки операционной системы в оперативную память каждая ее подсистема займет определенное место памяти (см. рис. 2.4). Затем процессор консольных команд выдает на экран приглашение в виде имени активного дисковода, символа > и курсора

A>_

Далее вы на клавиатуре набираете команду и через базовую систему ввода-вывода она поступает в буфер процессора консольных команд. Процессор консольных

команд анализирует и проверяет синтаксис поступившей команды, а затем решает, что делать дальше в зависимости от класса команд, так как каждая команда является либо резидентной, либо обслуживающей.

При поступлении резидентной команды процессор консольных команд выполняет ее немедленно, используя две другие подсистемы — базовую дисковую операционную систему и базовую систему ввода-вывода.

При вводе обслуживающей команды процессор консольных команд обращается к базовой дисковой операционной системе и базовой системе ввода-вывода, которые отыскивают на диске файл этой команды и загружают его в оперативную память начиная с адреса 0100H. Затем управление передается введенной обслуживающей команде, которая действует так же и на тех же правах, что и резидентная команда. После окончания обслуживающей команды управление передается на адрес 0000H зоны связи и происходит выход на процессор консольных команд, о чем известит вам появившаяся на экране подсказка A>.

Возможно, что вы ввели команду, которая не является ни резидентной, ни обслуживающей. Тогда процессор консольных команд выдает имя введенной команды и знак вопроса.

2.2.3. КАК ПОЛЬЗОВАТЬСЯ РЕЗИДЕНТНЫМИ КОМАНДАМИ?

Резидентные команды, часто называемые встроенными, являются частью операционной системы и находятся в той зоне памяти, которая отведена под операционную систему. После ввода резидентной команды процессор консольных команд проверяет ее синтаксис и при отсутствии ошибок немедленно приступает к ее выполнению.

Операционная система SCP располагает шестью резидентными командами, представленными в табл. 2.1. Наиболее употребительны команды DIR, ERA, REN, TYPE. Ниже приводится описание этих команд, а в табл. 2.2 — сообщения об ошибках, возникающих при их выполнении.

Резидентные команды

Таблица 2.1

Команда	Назначение команды
DIR	Вызов каталога диска
ERA	Стирание файлов
REN	Переименование файлов
TYPE	Вывод текстового файла с диска на экран
SAVE	Запись блоков оперативной памяти на диск в виде файла
USER	Установка зон пользователя на диске

Замечание! В строке ввода после команды могут следовать различные параметры. Необходимо отделить команду и параметры пробелом. При записи формата команды будут использоваться квадратные скобки [ ] для обозначения необязательных параметров.

Команда DIR

С помощью этой команды вы просмотрите каталог диска, в котором указаны имя файла и его тип. Сообщения об ошибках при выполнении этой команды приведены в табл. 2.2. Названия файлов с атрибутом SYS командой DIR не индицируются.

Формат: DIR [имя дисковода :] [полное имя файла]

Таблица 2.2

Сообщения об ошибках при выполнении резидентных команд

Текст сообщения	Перевод	Пояснения, рекомендации по устранению
	Команда DIR	
NOT FOUND	Не найден	На диске нет файла с заданным именем
NO FILE	Нет файла	Диск пустой
	Команда ERA	
SCPX ERR ON<A>: FILE R/O	Ошибка SCPX на диске A:, файл с атрибутом R/O	Указанный файл нельзя стирать и в него нельзя записывать. Для его стирания надо сначала изменить атрибут на R/W командой STAT
ALL (Y/N)?	Все (Да/нет)	Дополнительный запрос — все ли файлы надо стирать? Y — да, N — нет. Нажмите соответствующую клавишу
	Команда REN	
FILE EXISTS	Файл существует	Для переименования файла выберите другое имя, т. е. это имя уже имеется в каталоге
NO FILE	Нет файла	На диске нет файла со старым обозначением

Примеры.

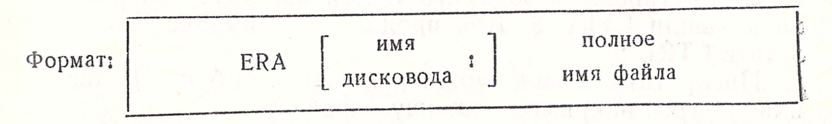
A>DIR. На экране появляется каталог диска, установленного в дисководе A.

A>DIR*.COM На экране появляются имена всех файлов типа COM с диска дисковода A.

A>DIR B: На экране появляется каталог диска, установленного в дисководе B.

Команда **ERA**

С помощью этой команды вы сотрете один или несколько файлов. Сообщения об ошибках приведены в табл. 2.2.



Формат:	ERA [имя дисковода :] полное имя файла

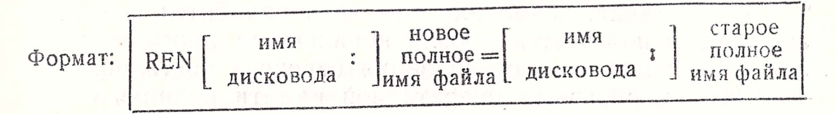
Примеры.

A> ERA ALFA.* Стираются все файлы с именем ALFA с диска в дисководе A.

A>ERA B:*.* Стираются все файлы с диска в дисководе B.

Команда **REN**

Иногда требуется изменить имя файла. Для этого используйте команду REN. Сообщения об ошибках приведены в табл. 2.2.



Формат:	REN [имя дисковода :] новое полное имя файла = [имя дисковода :] старое полное имя файла

В формате команды REN имя дисковода слева и справа от знака = должно быть одним и тем же.

Пример.

A >REN B:AAA.BBB=B:C.DDD На диске в дисководе B файл с именем C.DDD переименовывается в файл с именем AAA.BBB.

Команда **TYPE**

Этой командой вы воспользуетесь, когда захотите прочесть содержимое текстового файла. Вывод содержимого

любого другого типа файла не имеет смысла. Командой TYPE, указав полное имя текстового файла, вы получаете возможность вывести текст на экран.

Нажав одновременно клавиши **CTRL P**, вы обеспечите параллельную печать индицируемого текста на принтере.

Для останова выводимого текста нажмите одновременно клавиши **CTRL S**. Для продолжения вывода вновь нажмите **CTRL S**.

Прекратить вывод можно, нажав любую клавишу клавиатуры, например клавишу «пробел».

Формат:	TYPE	[имя дисковода :]	полное имя файла

Пример.

A > TYPE B:PRIMER.PRN Вывод содержимого файла печати с именем PRIMER.PRN с диска в дисководе В на экран.

Команда SAVE

Эта команда, возможно, вам не понадобится, в основном она используется профессиональными программистами. Ее назначение — организация переписи части области транзитных программ оперативной памяти начиная с адреса 0100Н (см. рис. 2.4) на диск в виде файла. Перепись осуществляется по сегментам объемом 256 байт каждый.

Формат:	SAVE	количество сегментов	[имя дисковода :]	полное имя файла

Пример.

A > SAVE 5 B:KOT.ABC

Из оперативной памяти на диск дисковода В в файл с именем KOT.ABC переписываются 5×256 байт оперативной памяти начиная с адреса 0100Н.

Команда USER

Эта команда, так же как и команда SAVE, используется нечасто. Воспользоваться ею вы можете в том случае, когда хотите один диск условно разбить на несколько независимых зон, и работать вы будете с этими зонами так же, как если бы это была не зона диска, а весь диск. Командой USER можно установить 16 независимых зон (0...15). После установки зоны все команды операционной системы обслуживают только эту зону. Когда команда USER не используется, вы работаете со всем диском, а по умолчанию принимается зона с номером 0. Необходимость в разбиении диска на зоны возникает тогда, когда с этим диском работают несколько пользователей или на нем хранятся файлы для совершенно не связанных функционально задач.

Формат:

USER	номер зоны

Пример.

A>USER 7 Установлена седьмая зона пользователя. Все вводимые в дальнейшем команды будут обслуживать только эту зону. Для работы в другой зоне необходимо установить командой USER и ее номер.

2.3. ОБСЛУЖИВАЮЩИЕ КОМАНДЫ — СРЕДСТВО РАСШИРЕНИЯ ВОЗМОЖНОСТЕЙ ОПЕРАЦИОННОЙ СИСТЕМЫ

Назначение

Обслуживающие программы расширяют возможности операционной системы и обеспечивают сервис работы пользователя на персональном компьютере. Обслуживающие программы, часто называемые транзитными или сервисными командами, не входят в состав операционной

системы, а хранятся в виде файлов типа COM на диске. Имя обслуживающей программы совпадает с именем этого файла. Операционная система обращается с файлом обслуживающей команды, как с обычным файлом: находит его на диске по имени и переписывает содержимое этого файла в область транзитных программ. Затем управление передается переписанной программе и начинается выполнение введенной команды.

Для персонального компьютера Роботрон 1715 создан целый ряд всевозможных обслуживающих команд, имеющих в большинстве своем уникальное назначение, хотя среди этого множества вы, возможно, и обнаружите дублирующие друг друга команды. Особенно широко используются обслуживающие команды, поставляемые вместе с аппаратной частью Роботрона 1715, и программа POWER, реализующая множество разнообразных команд сервиса.

В настоящем параграфе вы ознакомитесь с назначением и спецификой использования стандартного набора обслуживающих команд, а также получите краткий перечень сведений о назначении команд комплекса

Таблица 2.3

Обслуживающие команды

Команда	Назначение команды
DUMP	Индикация содержимого файла в кодах на экране
INIT	Форматирование диска
PIP	Копирование файлов
SGEN	Запись операционной системы на диск
STAT	Выдача информации о состоянии файлов и дисков, а также изменение этого состояния
SUBM	Запуск командного файла

Таблица 2.4

Сообщения при выполнении обслуживающих команд

Текст сообщения	Перевод	Пояснения, рекомендации
Команда DUMP		
ILLEGAL FILE NAME	Недопустимое имя файла	В команде либо не указано имя файла, либо оно многозначное
FILE NOT FOUND	Файл не найден	На диске нет файла с заданным именем
Команда INIT		
==> INVALID DRIVE	Неправильное имя дисковода	Вновь введите имя дисковода
==> INVALID FORMAT	Неправильный формат	Необходимо выбрать из меню другой формат
==> FORMATING COMPLETE	Форматирование закончено	После окончания форматирования вновь предлагается ввести имя дисковода и можно продолжить форматирование другого диска. Для окончания нажмите CTRL C.
ERROR ==> BAD TRACK	Ошибка ==> плохая дорожка	Дорожка, для которой появилось это сообщение, непригодна для хранения информации. Устранить этот дефект невозможно
ERROR ==> DRIVE	Ошибка ==> дис-	Форматируемый диск не вставлен

Текст сообщения	Перевод	Пояснения, рекомендации
NOT READY	ковод не готов	в дисковод. Вставьте его
ERROR ==> DISK WRITE PROTECTED	Ошибка ==> диск защищен от записи	У форматируемого диска имеется физическая защита записи. Для ее устранения нажмите клавишу ЕТ
ERROR ==> INCORRECT SCP — VERSION	Ошибка ==> некорректная версия SCP	Программа INIT не может работать в данной операционной системе. Надо поставить стандартную систему SCP
Команда PIP		
DISK READ ERROR	Ошибка считывания с диска	При считывании или записи обнаружена ошибка. Надо снова ввести команду
DISK WRITE ERROR	Ошибка записи на диск	
BAD PARAMETER	Неправильный параметр	Надо исправить параметр
INVALID FORMAT	Неправильный формат	Надо внести исправления в команду PIP
NO DIRECTORY SPACE	Нет места в каталоге	Невозможно внести в каталог копируемый файл
NO FILE	Нет файла	Файла с указанным именем не существует
START NOT FOUND	Не найде-	Не найдены строки

Текст сообщения	Перевод	Пояснения, рекомендации
	но начало	символов для ограниченного копирования
QUIT NOT FOUND	Не найдено окончание	
NOT FOUND	Не найден	Названный файл не найден
UNRECOGNIZED DESTINATION	Нераспознанный файл	Запрещенный файл назначения
CANNOT READ	Нельзя прочитать	Запрещенный или отсутствующий исходный файл
INVALID PIP FORMAT	Неправильный формат PIP	В команде отсутствует знак равенства или есть запрещенный пробел
INVALID SEPARATOR	Неправильный разделитель	В списке параметров используется запрещенный разделительный знак
Команда SGEN		
OUT OF MEMORY!	Выход за границу памяти	Превышение емкости оперативного буфера
DISK ERROR!	Ошибка диска	Ошибка записи или считывания на диске
OUT OF OFFSET!	Выход за границы смещения	Превышение емкости системных дорожек на диске
DIRECTORY FULL!	Каталог заполнен	Каталог диска заполнен полностью
DISK FULL!	Диск заполнен	Диск заполнен полностью

Текст сообщения	Перевод	Пояснения, рекомендации
LOADER GREATER THEN FILE!	Загрузчик превышает файл	Предостережение. Загруженный файл меньше, чем указанная в его описателе длина
SOURCE FILE NOT FOUND!	Файл-источник не найден	На диске нет файла с эталоном операционной системы
BOOT LOADER-TYPE FALSE!	Плохой загрузчик	На системных дорожках диска-источника нет достоверной SCP
DATA-PASSWORD FALSE!	Ложный пароль данных системы	Недостоверный идентификатор SCP
SYSTEM-PASSWORD FALSE!	—	Несовместимый идентификатор системы между SCP и SGEN
NO OFFSET!	—	На системных дорожках диска нет операционной системы SCP
DRAVE NOT EXIST!	Дисковод отсутствует	Нет привода с указанным именем. Задайте другое имя
Команда STAT		
FILE NOT FOUND	Файл не найден	Исправьте имя файла
INVALID FILE INDICATOR	Неправильные параметры файла	Надо исправить параметры файла (R/W, R/O, SYS, DIR)
INVALID DISK ASSIGNMENT	Неправильное на-	Неправильное распределение атрибутов

Текст сообщения	Перевод	Пояснения, рекомендации
	значение диска	для диска
BAD DELIMITER	Неправильный разделитель	Надо исправить разделители
INVALID ASSIGNMENT	Неправильное назначение	Ошибочное распределение канала

Команда SUBM

Текст сообщения	Перевод	Пояснения, рекомендации
NO 'SUB' FILE PRESENT:	Файла "... SUB" не существует	Нет командного файла
PARAMETER ERROR:	Ошибка параметров	Имеется ошибка в параметрах команды в командном файле
COMMAND TOO LONG:	Длинная команда	Длина строки с командой превышает 126 символов
COMMAND BUFFER OVERF LOW:	Переполнение буфера команды	Длина командного файла превышает 1023 символа
DIRECTORY FULL:	Каталог заполнен	Для командного файла нет свободного места в каталоге
DISK WRITE ERROR:	Ошибка записи на диск	Аппаратная ошибка в дисководе
INVALID CONTROL CHARACTER:	Неправильный управляющий символ	В командном файле использован ошибочный управляющий символ

POWER. Шесть обслуживающих команд стандартного набора приведены в табл. 2.3. Возникающие при их выполнении сообщения о дальнейших действиях или ошибках сведены в табл. 2.4.

Для ввода команды наберите имя файла, который ее реализует, без указания типа.

Обратите внимание!

●●● В строке ввода для отделения команды от параметров обязательно используйте пробел.

●●● При записи формата команды для указания необязательных параметров используются квадратные скобки.

●●● Полное имя файла состоит из имени файла и типа, разделенных точкой.

●●● Для выхода из обслуживающей программы нажмите одновременно две клавиши **CTRL C.**

Команда DUMP

Этой командой вряд ли вы будете пользоваться, так как ее назначение — индикация содержимого файла, представленного в виде кодов. На экран выдается таблица кодов. При выполнении команды возможны ошибки, сообщения о которых представлены в табл. 2.4.

Формат:	DUMP	полное имя файла

Пример.

A >DUMP KBOM.AB На экран выдается таблица из 17 столбцов с кодами.

Команда INIT

Из-за частого использования команды INIT рекомендуем отнестись к изложенному ниже материалу более внимательно и освоить процедуру работы с ней. Команда INIT предназначена для форматирования диска, т. е. за-

дания ему формата, необходимого для работы в конкретной модели дисковода. Отсутствие или неправильное форматирование диска делает невозможной работу с ним. Форматируемый диск может находиться в любом дисководе. Команда INIT выполняется в диалоге — вам необходимо ответить на ряд вопросов, после чего начнется форматирование. При его успешном окончании будет выдано сообщение FORMATING COMPLETE (форматиро-

Таблица 25.

Диалог между INIT и пользователем

Сообщение	Перевод сообщения	Действия пользователя
PLEASE ENTER DRIVE: A	Укажите, пожалуйста, дисковод: A	B
DISK FORMAT CAN BE: 0=DD-SS 16*256*40 1=DD-SS 26*128*40 2=DD-SS 5*1024*40	Форматы диска могут быть следующими: . . .	
PLEASE SELECT FORMAT: Ø	Выберите, пожалуйста, формат: 0	2
WARNING→ALL FILES WILL BE SCRATCHED (Y/N):	Предупреждение. Все файлы будут уничтожены (д/н):	Y
FORMATING TRACK (01)	Форматируется дорожка (номер)	Вы наблюдаете за номером форматируемой дорожки
FORMATING COMPLETE	Форматирование закончено	Нажмите клавиши CTRL C или форматируйте следующий диск

вание окончено) и предложено провести форматирование другого диска. Для выхода из команды INIT нажмите одновременно клавиши **CTRL C**.

Формат:	INIT

Пример диалога между INIT и пользователем приведен в табл. 2.5. В этом диалоге предполагается, что диск будет использован в дисководе типа MFS 1.2. При форматировании чистый диск установлен в дисководе B.

При появлении сообщений об ошибках в процессе выполнения команды INIT обратитесь за разъяснением к табл. 2.4.

Команда **PIP**

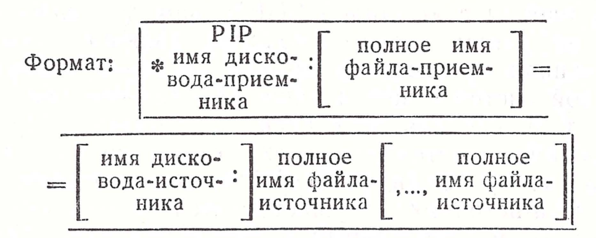
Главная функция команды PIP — копирование файлов с одного диска на другой. Помимо этого вам предоставляется возможность копирования участков данных, слияния в один файл нескольких файлов, печати содержимого файлов в специальном формате. Предусмотрена также возможность с помощью 17 параметров модифицировать функции команды PIP.

Здесь вы ознакомитесь с процедурой копирования и слияния файлов. Для более глубокого изучения возможностей команды PIP, определяемых ее параметрами, вам следует обратиться к техническому описанию [19].

Обратите внимание! Если вы располагаете программой POWER, то более удобной по сравнению с командой PIP является аналогичная команда копирования COPY.

При записи формата используется понятие файла (дисковода)-приемника для указания места, куда производится копирование, и файла (дисковода)-источника для

указания места, откуда производится копирование. При копировании надо обязательно указывать имя дисковода-приемника и имя файла-источника.



Вводится команда PIP двумя строками. В первой строке вы набираете слово PIP и, нажав клавишу **ЕТ**, вводите его. В ответ на экране во второй строке появляется символ *, после которого вы набираете строку с указанием дисководов и файлов копирования и нажимаете клавишу **ЕТ**. Начинается процесс копирования. После его окончания на экране вновь появляется символ * и вы можете указать следующую группу копирования и т. д. Для выхода из команды PIP нажмите одновременно две клавиши **CTRL C**.

При появлении сообщений во время выполнения команды копирования обратитесь к табл. 2.4.

Примеры.

◆ B:=*.* Копирование всех файлов с диска дисковода A (по умолчанию) на диск дисковода B.

◆ A:=B:*.* Копирование всех файлов с диска дисковода B на диск дисковода A.

◆ B:=KROT.* Копирование всех типов файлов с именем KROT с диска дисковода A (по умолчанию) на диск дисковода B.

◆ B:LOT.D=RAK.C Копирование файла RAK.C с диска дисковода A (по умолчанию) с переименованием его в LOT.D на диск дисковода B.

◆ B:B1.ABC=A1.C,A2.C,A3.C Копирование и слияние трех файлов с диска дисковода A в один файл B1.ABC на диск дисковода B. Файлы сливаются в указанной последовательности один за другим.

Команда SGEN

Эта команда позволит вам записать операционную систему на системные дорожки диска или в виде файла.

Для работы с командой SGEN вам понадобятся два диска: на одном диске—источнике должна находиться операционная система; на втором диске—приемнике операционной системы нет, и ваша задача — ее там сформировать, т. е. сделать этот диск системным. Процедура формирования системного диска организована в виде диалога между командой SGEN и пользователем. Вначале с диска-источника осуществляется запись операционной системы в оперативный буфер с помощью меню, где предлагаются три возможных варианта записи. После ввода номера варианта на экране произойдет смена меню и появится второе меню, где наряду с уже известными вариантами предлагаются еще четыре новых для организации записи операционной системы из оперативного буфера на диск-приемник. Вам следует ввести номер варианта записи (4—7), и начнется формирование системного диска.

Формат: SGEN

●●● Меню записи операционной системы с диска-источника в оперативный буфер:

1 — LOAD COM-FILE SYSTEM FROM DISK (загрузка из файла в оперативный буфер);

2 — LOAD ACTIVE SYSTEM FROM MEMORY (загрузка операционной системы из оперативной памяти в оперативный буфер);

3 — LOAD BOOT SYSTEM FROM DISK (загрузка с системных дорожек в оперативный буфер);

4 — EXIT (завершение команды SGEN).

●●● Меню записи операционной системы из оперативного буфера на диск-приемник:

1 — LOAD COM-FILE SYSTEM FROM DISK 2 — LOAD ACTIVE SYSTEM FROM MEMORY 3 — LOAD BOOT SYSTEM FROM DISK	из предыдущего меню, эти варианты здесь не используются;

4 — RESET DISK (режим смены дисков);

5 — SAVE COM-FILE SYSTEM TO DISK (запись операционной системы на диск в виде файла);

6 — SAVE ACTIVE SYSTEM TO MEMORY (загрузка операционной системы из оперативного буфера в оперативную память);

7 — SAVE BOOT SYSTEM TO DISK (запись операционной системы на системные дорожки диска);

8 — EXIT (завершение команды SGEN).

При выполнении команды SGEN возможно появление сообщений об ошибках. Понять их и произвести соответствующие действия вам поможет табл. 2.4.

Команда STAT

Этой командой вы будете пользоваться для получения информации о состоянии диска и файла, а также для изменения этого состояния. Необходимым условием выполнения этой команды является наличие системного диска в дисководе A. При появлении сообщений во время выполнения команды STAT обратитесь к табл. 2.4. Команда STAT выполняет различные функции, которые определяются параметрами. Рассмотрим их.

Формат:
```
STAT
```

На экране появляется сообщение об объеме (в килобайтах) свободной зоны на диске:

A: R/W, SPACE: * K**

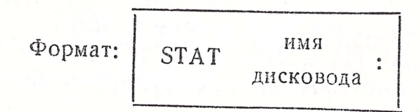
атрибут (R/W) — объем свободной зоны (***)

Если диски установлены в обоих дисководах, то выводится сообщение о свободных зонах для этих дисков.

Пример.

A>STAT

A:R/W, SPACE: 23K



На экране появляется сообщение об объеме (в килобайтах) свободной зоны на диске в указанном дисководе:

BYTES REMANING ON имя дисковода :*** K

(*** — объем свободной зоны)

Пример.

A>STAT B:

BYTES REMANING ON B: 134K

Формат: STAT [имя дисковода :] полное имя файла

На экране появляется информация о структуре файла:

RECS	**BYTES**	**EXT**	**ACC**	имя дисковода:
***	** K	*	R/*	полное имя файла

BYTES REMANING ON имя дисковода **: *** K**

(*** — объем свободной зоны)

В сообщении вместо символов * стоят конкретные числа, которые указывают следующее:

в первой колонке (RECS) — число записей по 128 байт в файле;

во второй колонке (BYTES) — объем файла в килобайтах;

в третьей колонке (EXT) — число зон по 32 Кбайт в файле;

в четвертой колонке (ACC) — атрибут файла (R/W, R/O).

Пример вызова информации о структуре файлов типа COM.

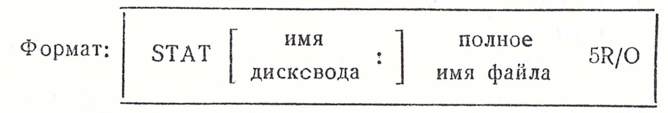
```
A >STAT *.COM
     RECS      BYTES       EXT        ACC
       97        14K         1        R/O          A:INIT.COM
       58         8K         1        R/O          A:PIP.COM
       42         6K         1        R/O          A:STAT.COM
   BYTES REMAINING ON      A:24K
```

Формат: STAT [имя дисковода:] = R/O

На весь диск устанавливается атрибут R/O, т. е. с диска можно только читать, запись на него запрещена. Этот режим сохраняется лишь на время работы с данным диском. После нового запуска диск вновь приобретает атрибут R/W и становятся возможными и запись, и чтение.

Пример.

A>STAT B:=R/O



Установка атрибута R/O. Указанный файл становится недоступным для записи, возможно только чтение. По окончании команды выдается сообщение:

полное имя файла SET TO R/O

Пример.
A>STAT AA.COM 5R/O
AA.COM SET TO R/O
Файл AA.COM становится недоступным для чтения.

Формат:	STAT	[имя дисковода	:]	полное имя файла	&R/W

Установка атрибута R/W. В файле можно производить и запись, и чтение. По окончании команды выдается сообщение:

полное имя файла SET TO R/W

Пример.
A>STAT ZELO.B &/RW
ZELO.B SET TO R/W

Формат:	STAT	[имя дисковода	:]	полное имя файла	¤ SYS

Присваивание файлу атрибута SYS. Файлы, имеющие атрибут SYS, выдают на индикацию свои имена в каталоге по команде DIR. По окончании команды выдается сообщение:

полное имя файла SET TO SYS

Пример.
A>STAT A.A ¤ SYS
A.A SET TO SYS

Формат:	STAT	[имя дисковода	:]	полное имя файла	¤ DIR

Присваивание файлу атрибута DIR. Подобные файлы получают возможность индицировать свои имена в ката-

логе по команде DIR. По окончании команды выдается сообщение:

полное имя файла SET TO DIR

Формат:	STAT [имя дисковода :] DSK:

Характеристика диска. Понять сообщение, которое выдается в результате действия этой команды, предлагаем вам на примере:

A>STAT DSK:

Сообщение	Пояснение
A: DRIVE CHARACTERISTICS	Состояние диска в дисководе А
1472: 128 BYTE RECORD CAPACITY 184: KILOBYTE DRIVE CAPACITY	Емкость диска 1472 записи по 128 байт, что составляет 184 Кбайт
64: 32 BYTE DIRECTORY ENTRIES 64: CHECKED DIRECTORY ENTRIES	Каталог вмещает 64 описателя файла по 32 байт. т. е. объем каталога равен 2 Кбайт
256: RECORDS/EXTENT	Каждый описатель каталога рассчитан на максимальную длину файла из 256 записей (32 Кбайт)
16: RECORDS/BLOCK	В каждом блоке содержатся 16 записей по 128 байт, что составляет 2 Кбайт
40: SECTORS/TRACK	На каждой дорожке 40 записей по 128 байт
3: RESERVED TRACKS	Три резервные дорожки (0, 1, 2) для записи операционной системы

В связи с тем что параметры VAL, USR, DEV команды STAT используются редко, здесь они не рассматриваются.

Команда SUBM

С помощью команды SUBM вы запустите командный файл в работу.

Командный файл — совокупность вызовов резидентных и обслуживающих команд операционной системы, записанная на диске в виде файла. В момент выполнения командного файла надо обеспечить наличие на диске используемых в нем обслуживающих команд, в противном случае его выполнение становится невозможным. Командный файл целесообразно применять в том случае, когда требуется многократное повторение одних и тех же действий, а вы хотите себя избавить от подобной процедуры.

Как сформировать командный файл? Для этого вы должны научиться работать с текстовым редактором. Это достаточно легко сделать, если вы обратитесь к гл. 3, где представлен один из вариантов текстового редактора РЕФОР. Работая в среде пакета прикладных программ РЕФОР, вы создадите текстовый файл, состоящий из команд, присвоите ему любое имя с типом SUB и запишете на диск, с которым будете в дальнейшем работать.

Пример 1 командного файла PRIMER1. SUB.

ERA B:*.TBL — стирание всех файлов типа TBL

DIR
DIR B: } индикация каталогов обоих дисков

PIP B:=BEP.* — копирование на диск дисковода B всех файлов с именем BEP с диска дисковода A

Набор каждой строки командного файла должен завершаться нажатием клавиши ET.

Вместо конкретных указаний полных имен файлов в команде можете использовать параметры, т. е. символ ¤ и порядковый номер этого параметра. Нумерацию

следует проводить в порядке применения этих файлов. Необходимость в этом возникает тогда, когда имена файлов определяются конкретной ситуацией. В этом случае имена файлов вы записываете в виде параметров команды SUBM. Рассмотрим, как изменится командный файл PRIMER1. SUB при использовании в нем вместо конкретных имен параметров ¤ 1, ¤ 2.

Пример 2 командного файла PRIMER2.SUB.

ERA B:*. ¤ 1 — тип TBL заменен ¤ 1
DIR
DIR B:
PIP B:= ¤ 2.* — имя BEP заменено ¤ 2

Для запуска сформированного командного файла следует вернуться в среду операционной системы и применить команду SUBM.

Формат:	SUBM	имя командного файла	[параметры, разделенные пробелом]

В команде SUBM не указывается тип командного файла. В качестве параметров используются имена файлов, с которыми будет работать командный файл. Порядок следования параметров соответствует порядку следования их в командном файле.

Для примера 1 команда запуска имеет вид:

A> SUBM PRIMER1

Для примера 2 в предположении, что используются файлы примера 1, команда запуска имеет вид:

A >SUBM PRIMER2 TBL BEP

При выполнении командного файла PRIMER2 параметр ¤ 1 примет значение TBL, параметр ¤ 2 — значение BEP. В результате вы получите тот же командный файл, что и в примере 1.

Изменение параметров команды SUBM приведет к изменению имени файла, а значит, и команды.

Пример 3.

Введя команду

A > SUBM PRIMER2 MAC AAA,

вы произведете запуск командного файла PRIMER2 для отличных от примера 1 файлов:

ERA B:*.MAC — параметр для типа ¤ 1 заменен на тип MAC
DIR
DIR B:
PIP B:AAA.* — параметр имени ¤ 2 заменен на имя AAA

Для разъяснения сообщений, возникающих в процессе выполнения команды SUBM, обратитесь к табл. 2.4.

Программа POWER

Приобретя программу POWER, вы существенно расширите свои возможности при работе в среде операционной системы. Однако это потребуется вам только в том случае, если вы приобрели навыки работы на компьютере Роботрон 1715 и хотите совершенствоваться дальше. Здесь мы познакомим вас только с перечнем команд POWER, чтобы вы имели представление о нем. Для освоения этих команд загрузите POWER, после появления на экране подсказки в виде AO= нажмите клавишу ? и получите подробное описание команд.

CHECK — подсчет контрольных сумм указанных файлов
CM — сравнение двух областей памяти
COPY — копирование файлов
DIR — оглавление диска
DISK — информация об организации диска
DS — просмотр и заполнение памяти
DUMP — вывод текстового массива
ERA — стирание файлов
EXIT — выход в операционную систему
FILL — заполнение памяти константой BYTE
GO — запуск программы
GROUP — список блоков данного файла

JP (EX) — передача управления по адресу

LOAD (SAVE) — загрузка (выгрузка) файла

LOG — информация о состоянии POWER

MOVE — перемещение области памяти

READ (WRITE) — прямой доступ к диску при чтении (записи)

READGR (WRITEGR) — обмен с диском

RECLAIM — восстановление стертых файлов

REN — переименование

RESET — сброс диска

RUN — загрузка и передача управления другой программе

SEARCH — поиск символа, байта, текста в памяти

SET — установка старшего бита в коде символа имени или типа файла

SETDIR (SETSYS, SETRO, SETWR) — установка атрибутов файла

SIZE — определение размеров файлов

SORT — сортировка каталога

SPEED — установка скорости вывода на дисплей

STAT — информация о заполненности диска

TEST — тестирование диска

TYPE (TYPEX, TYPEH, TYPEA) — вывод текстового файла на дисплей

USER (XUSER) — использование областей пользователя

2.4. ПРОВЕРЯЕМ СЕБЯ. РАЗОБРАЛИСЬ ЛИ ВЫ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ?

Упражнения

2.1. Что следует понимать под программным обеспечением персонального компьютера, в том числе и Роботрона 1715?

2.2. Что такое файл?

2.3. Как различаются файлы между собой?

2.4. В чем суть файловой системы, лежащей в основе организации операционной системы?

2.5. Из каких трех подсистем состоит операционная система?

2.6. Назначение процессора консольных команд.

2.7. Назначение базовой дисковой операционной системы.

2.8. Структура базовой системы ввода-вывода.

2.9. В чем заключается различие между резидентными и обслуживающими командами?

2.10. Правило ввода команд.

2.11. Как выводится каталог диска?

2.12. Как вывести содержимое текстового файла?

2.13. Перечислите резидентные команды.

2.14. Вы хотите переписать программы с одного диска на другой. Какая при этом должна быть последовательность действий?

2.15. Какая команда записывает операционную систему на новый диск?

2.16. Каковы функции команды STAT?

Ответы к упражнениям

2.1. Состав программного обеспечения:

◆ системные программные средства, в состав которых входят:

операционная система, управляющая процессом обработки информации;

обслуживающие команды, которые предоставляют всевозможный сервис пользователю для работы в среде операционной системы;

программы контроля и диагностики работы аппаратной части;

◆ система программирования, состоящая из трансляторов и языков программирования;

◆ прикладные программные средства в виде пакетов прикладных программ и программ пользователя.

2.2. Совокупность логически связанных записей, занимающую определенное место во внешней памяти, называют файлом. Файлом представляются программы, данные.

2.3. Каждый файл имеет имя и тип, составляющие его полное имя. Имя каждого файла уникально и состоит не более чем из восьми символов. Тип файла зависит от используемых программ и состоит не более чем из трех символов. Имя и тип разделяются точкой.

При образовании полного имени нельзя использовать символы . ? *

2.4. Файловая система управляет распределением пространства диска под файлы. К файлу организован доступ только через каталог, где информация о нем хранится в описателе в виде полного имени и номеров занимаемых им блоков.

2.5. Процессор консольных команд, базовая дисковая операционная система, базовая система ввода-вывода.

2.6. Организация связи между пользователем и компьютером. Процессор консольных команд анализирует вводимые команды и при необходимости подключает базовую дисковую операционную систему, а также запускает прикладные программы.

2.7. Основная функция — организация доступа к файловой системе.

2.8. Базовая система ввода-вывода состоит из драйверов — программ, организующих обмен информацией между центральной частью и конкретным типом внешнего устройства. Необходимым условием управления работой внешнего устройства является соответствие драйвера и контроллера.

2.9. Резидентные команды входят в состав операционной системы. После ввода резидентной команды процессор консольных команд немедленно ее выполняет. Обслуживающие команды хранятся в виде файлов на дисках

и расширяют возможности операционной системы. После ввода обслуживающей команды процессор консольных команд подключает базовую операционную систему и базовую систему ввода-вывода, а по окончании загрузки соответствующего файла в оперативную память начинает ее выполнение.

2.10. После появления подсказки наберите имя команды и нажмите клавишу ввода **ET**. Если в команде имеются параметры, например имя файла, то команды и параметры при наборе разделяются пробелом.

2.11. Командой DIR.

2.12 Команда TYPE предназначена для вывода текстового файла на экран. Для синхронного вывода текста на принтер и экран следует нажать перед вводом команды TYPE клавиши **CTRL P**.

2.13. DIR, TYPE, ERA, REN, USER, SAVE.

2.14. Последовательность действий такова:

- загрузите операционную систему с системного диска в оперативную память и обеспечьте наличие на диске команд INIT, PIP;
- вставьте новый диск в любой дисковод;
- проведите форматирование нового диска обслуживающей командой INIT;
- введите команду PIP или загрузите в оперативную память программу POWER; поставьте в дисковод диск, с которого копируются файлы, предварительно вынув системный диск;
- проведите копирование с помощью команды PIP или, если вы работаете в среде программы POWER, то командой COPY.

2.15. Команда SGEN записывает операционную систему на системные дорожки диска 0—2 или в виде файла.

2.16. Функции команды STAT таковы:

- справка о свободной зоне на диске в килобайтах;
- справка о структуре файла: число записей, объем фай-

ла (в килобайтах), число зон, атрибут файла (R/W, R/O);

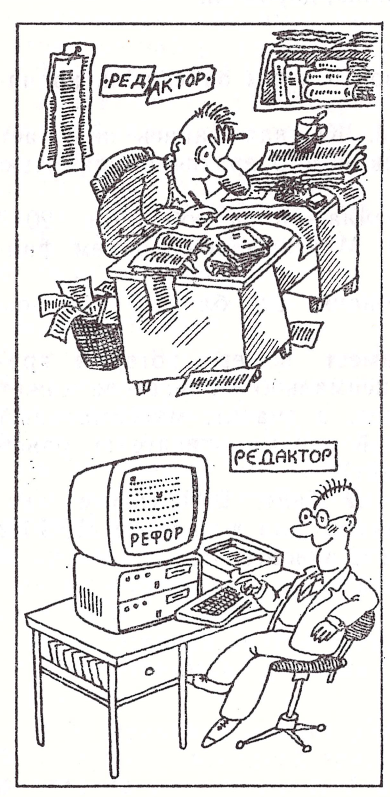
- установка на файл атрибутов: R/O (разрешено только чтение); R/W (разрешены и чтение, и запись); SYS (запрет индикации имени файла в каталоге); DIR (разрешена индикация имени файла в каталоге);
- справка о структуре диска в виде сообщения о емкости диска, каталога, описателя, блока, дорожки.

Справка.

- Доступ к файлам организован через описатели, хранящиеся в каталоге.
- Файл состоит из записей. Каждая запись занимает 128 байт. Обмен между диском и оперативной памятью производится по записям.
- Минимальный объем файла — блок объемом 2048 байт=2 Кбайт (16 записей). Максимальный объем файла — весь диск.
- Записи файла сгруппированы по блокам объемом 2 Кбайт (16 записей).
- Каждый блок файла имеет номер, который хранится в описателе файла. Максимально в описателе может находиться 16 номеров блоков, а значит, максимальный объем файла, соответствующий одному описателю, равен 32 Кбайт (16 блоков по 2 Кбайт).
- Объем описателя файла 32 байт. В каталоге максимально размещаются, например, для дисковода MFS 1.2 64 описателя, для MFS 1.4 — 128 описателей.

3

ПРИМЕНЯЕМ РЕФОР — СИСТЕМУ ПОДГОТОВКИ ТЕКСТОВ



3.1. ЗНАКОМСТВО С РЕФОРом

Назначение

РЕФОР (РЕдактор — ФОРматтер) [15] является пакетом прикладных программ, основное назначение которого состоит в подготовке разнообразных текстовых материалов и документов в экономической, торговой, административно-хозяйственной, редакционно-издательской и других подобных видах человеческой деятельности. С помощью РЕФОРа вы также можете формировать тексты программ на алгоритмических языках, например Паскале.

Если сравнить РЕФОР с аналогичными пакетами программ для других компьютеров, например с EDI, TED, DOC для ЭВМ типа СМ-4 или ED для персо-

нального компьютера типа ДВК, то следует отметить, что РЕФОР обладает значительно более развитыми средствами подготовки, хранения и печати текстовых документов.

Объем пакета РЕФОР составляет 84 Кбайт.

Основные понятия

Переходя к рассмотрению средств РЕФОРа, договоримся вначале о том, что в дальнейшем будем понимать под редактированием и форматированием текстов. В редактирование включаются операции пользователя, связанные как с начальным формированием текста посредством ввода с клавиатуры, так и с внесением в уже имеющийся текст разнообразных изменений (вставки, удаления и замены отдельных символов, слов и целых текстовых фрагментов). Под форматированием будем понимать операции, определяющие структуру подготавливаемого текста, например выравнивание строк по границе текста, центрирование заголовков, распределение текста по отдельным страницам документа, формирование таблиц, установку длины строки и др.

РЕФОР позволяет подготавливать тексты на русском языке с использованием прописных и строчных букв, а также с включением в текст прописных латинских букв. Работая с РЕФОРом, вы можете выполнять следующие операции:

- ввод текста с клавиатуры компьютера с автоматическим формированием строк заданного размера;
- формирование таблиц с использованием табуляции, аналогично применяемой при работе на пишущих машинках;
- редактирование текста, состоящее в замене, вставке, удалении, дублировании или перемещении текстовых фрагментов;
- форматирование текста при его подготовке к печати

с возможностью выделения заголовков, установки размеров строк и страниц, выбора шрифтов и способов печати;
◆ запись текста в файл на магнитном диске, копирование, переименование и удаление файлов;
◆ печать текста на бумаге в виде рулона или отдельных страниц с возможностью вставки данных из другого файла или вводимых с клавиатуры.

Как управлять работой РЕФОРа?

Управление работой РЕФОРа производится посредством ввода с клавиатуры буквенных команд. Команды вводятся путем нажатия одной или последовательно нескольких клавиш в любом регистре. Причем в ряде случаев ввод букв должен сопровождаться нажатием клавиши **CTRL**, что в дальнейшем будет обозначаться кавычкой (") перед соответствующей буквой. Часть команд может быть введена нажатием одной из функциональных клавиш, дублирующих отдельные буквенные команды.

Все команды РЕФОРа сгруппированы и представлены в виде нескольких меню, схема взаимодействия которых в виде направленного графа изображена на рис. 3.1. Меню — это перечень возможных вариантов ваших действий в данной ситуации с указанием по каждому варианту названия клавиш, которые следует нажать.

Роль системного меню

При входе в РЕФОР первоначально вы всегда окажетесь в системном меню, название которого связано с возможностью выполнения в этом меню ряда операций, свойственных операционной системе SCP: открытие файлов, их копирование, переименование, печать и удаление, вывод каталога файлов на гибком магнитном диске, запуск отдельных программ операционной системы (INIT — для разметки магнитных дисков, STAT — для оценки пространства на диске и др.).

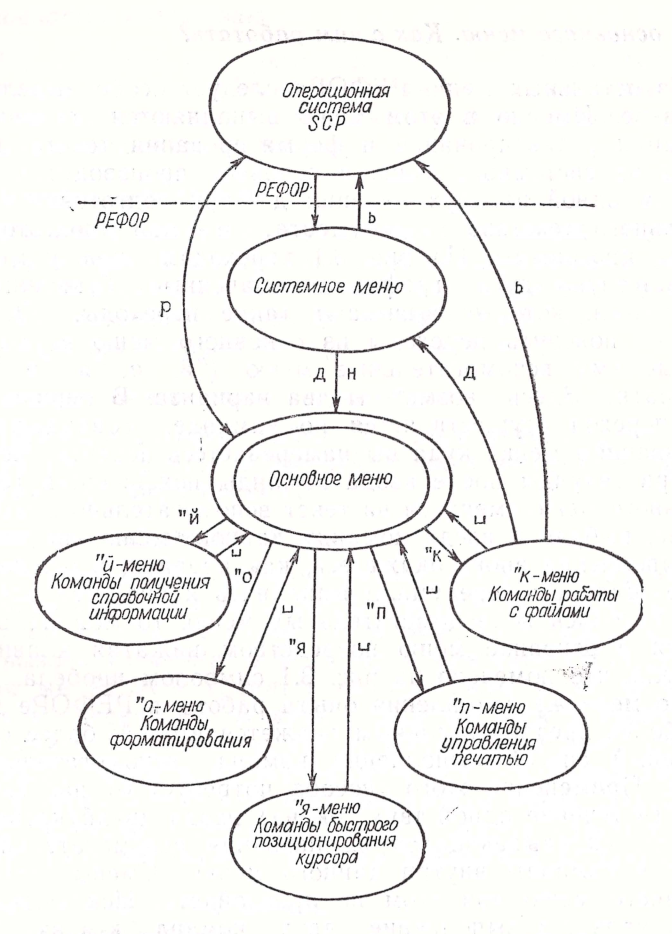


Рис. 3.1. Схема взаимодействия меню РЕФОРа

Роль основного меню. Как с ним работать?

Из остальных меню РЕФОРа следует особо выделить основное. Именно в этом меню выполняются основные операции редактирования и форматирования текста. Переход из системного меню в основное производится нажатием одной из двух клавиш: **д** — при подготовке произвольного текстового документа; **н** — при подготовке текста программы. На рис. 3.1 переходам между меню соответствуют дуги графа, обозначенные буквенными командами, которые вызывают такие переходы. Особо следует пояснить переходы из основного меню в подчиненные ему вспомогательные меню (**″й**, **″о**, **″я**, **″п**, **″к**) и обратно. Здесь возможны два варианта. В первом из них переход осуществляется по команде, совпадающей с названием меню, куда вы намереваетесь перейти. Через две-три секунды после ввода команды поясняющий текст основного меню сменится на текст вспомогательного меню. Далее, выбрав и введя команду данного меню, вы после ее выполнения вновь окажетесь, как правило, в основном меню. Если вы передумали выполнять какие-либо операции, находясь во вспомогательных меню, то можете вернуться в основное меню посредством нажатия клавиши пробела, что помечено на рис. 3.1 символом пробела.

По мере же накопления опыта работы в РЕФОРе для вас более предпочтительным окажется второй, более оперативный способ выполнения команд вспомогательных меню. Применение этого способа потребует от вас быстрого (в течение одной-двух секунд) ввода двухбуквенных команд, складывающихся из названия вспомогательного меню и команды внутри данного меню. Смены текста основного меню при этом не произойдет. Исключением здесь является выполнение двух команд **″к**-меню: **″кд** и **″кь**. По их окончании вы окажетесь соответственно в системном меню или в операционной системе SCP.

Руководство к действию

Ну а теперь вы приглашаетесь на ознакомительную прогулку по меню РЕФОРа. Постарайтесь при этом следовать нашим советам. Все последующие действия рекомендуем выполнять по шагам.

Подготовьте компьютер

Включите питание компьютера клавишей **POWER**, вставьте гибкий магнитный диск с операционной системой SCP в дисковод А и закройте его защелку. Произойдет автоматическая загрузка операционной системы, подтверждаемая выводом на экран подсказки **A>**.

Установите режим ввода русских букв

Нажмите клавишу **SI/SO**. При этом индикаторная лампочка рядом с этой клавишей должна загореться. Далее нажмите клавишу пробела.

Проверьте содержимое диска

Уясните, каким образом размещены на дисках операционная система и пакет программ РЕФОР. Если РЕФОР записан на том же диске, что и SCP, то наберите команду вызова РЕФОРа: а> рефор и нажмите клавишу **ЕТ**. Если же РЕФОР хранится на отдельном диске, то замените в дисководе А диск с SCP на диск с РЕФОРом и введите такую же команду. Начнется загрузка пакета РЕФОР. Вначале на экране появится рекламная заставка, а через 20—25 с — текст системного меню. Для быстрого перехода в системное меню нажмите клавишу пробела сразу же после появления рекламной заставки.

Вставьте при необходимости дополнительный диск

Примите решение, где будет фиксироваться в виде файла подготавливаемый текст. Если для хранения текс-

тового файла вами выделен отдельный диск, то вставьте его в дисковод В и закройте защелку, после чего введите команду системного меню **л**.

Установите уровень помощи

Находясь в системном меню, вы можете установить желаемый уровень вывода справочной информации РЕФОРа с помощью команды **х**. Рекомендуем вам для начала в ответ на запрос РЕФОРа ввести значение **3**, что соответствует уровню вывода максимального объема справочной информации во всех меню. На остальных уровнях — 1-м и 2-м объем справочной информации уменьшен.

Перейдите в основное меню

Введите команду системного меню **д**, что соответствует якобы вашему желанию создать текстовой документ. В ответ на соответствующий запрос РЕФОРа введите в качестве имени создаваемого текстового файла последовательность не более чем из восьми символов, например **прогулка**. Завершив ввод нажатием клавиши **ЕТ**, вы перейдете в основное меню. Знакомство со структурой информации, отображаемой на экране в этом меню, у вас еще впереди.

Воспользуйтесь возможностью самостоятельного обучения

Обучение основным командам РЕФОРа может быть организовано вами самостоятельно с помощью поясняющих текстов **″й**-меню. Согласно рис. 3.1 путь к этому меню лежит через основное меню, в котором вы сейчас находитесь. Введите команду **″й**, и вы окажетесь в меню помощи **″й**. Команды этого меню позволят вам просмотреть разнообразные справочные материалы:

- **б** пояснит использование команды форматирования строк **″б** основного меню;
- **ф** расшифрует назначение служебных символов в пра-

вой крайней колонке экрана (это сделаем и мы несколько ниже);

◆ **д** пояснит использование команд ″**п**-меню для управления печатью текстового документа;

◆ **с** расшифрует содержание верхней служебной строки основного меню;

◆ **м** пояснит способ установки левой и правой границ формируемого текста, межстрочного интервала, применение табуляции и приемов формирования таблиц;

◆ **п** пояснит правила внесения в текст специальных символов, так называемых маркеров начала и конца выделяемого текстового фрагмента — блока;

◆ **ж** пояснит правила перемещения и дублирования текстовых блоков;

◆ **и** выведет перечень наиболее часто используемых команд ввода текста.

Обучающие тексты ″**й**-меню, как правило, выводятся на экран по частям. Для их чередования согласно подсказке на экране нажимайте клавишу пробела. По окончании вывода обучающего текста вы автоматически вернетесь в основное меню. Поэтому для продолжения обучения вновь введите команду ″**й** и букву интересующего вас раздела справочной информации.

Прервать процесс обучения с возвращением в основное меню вы сможете в любой момент вводом команды прерывания ″**у** и последующим нажатием клавиши **ESC**.

Познакомьтесь с остальными меню

Продолжая ознакомительную прогулку по РЕФОРу, вы можете заглянуть и в остальные вспомогательные меню: ″**о**, ″**я**, ″**п** и ″**к**. Просмотр поясняющих текстов меню завершайте нажатием клавиши пробела, возвращающим вас каждый раз в основное меню.

Закончите работу

Завершите ознакомление с РЕФОРом вводом команды ″**кь**, которая обеспечит выход в операционную систе-

му SCP. Следующая встреча с РЕФОРом потребует от вас более активных и созидательных действий по вводу, редактированию и форматированию текстов.

3.2. ГОТОВИМ ТЕКСТ ПОЗДРАВИТЕЛЬНОЙ ОТКРЫТКИ

Изобилие средств РЕФОРа, с которыми вы ознакомились, просматривая текстовые пояснения в его различных меню, вызывает естественные затруднения при начальном освоении всего арсенала этих средств. Поэтому мы предлагаем вам пошаговое выполнение простых операций ввода, коррекции и печати текстов небольшого объема, например поздравительных открыток по случаю какого-либо праздника. Для выполнения таких операций вам потребуется лишь ограниченный перечень команд РЕФОРа.

Итак, в канун Нового года вы готовите тексты поздравительных посланий вашим друзьям и коллегам по работе. В вашем распоряжении пачка незаполненных почтовых открыток и компьютер Роботрон 1715, оснащенный пакетом РЕФОР.

Для осуществления задуманного вам не потребуются большие объемы памяти, в том числе и на гибком магнитном диске. Поэтому для простоты предположим, что пакет программ РЕФОР соседствует на диске с операционной системой SCP, а свободная его часть будет использована для запоминания подготавливаемых текстов поздравлений.

Все последующие ваши действия рекомендуем разбить на отдельные шаги.

Подготовьте компьютер

Включите питание компьютера и вставьте гибкий магнитный диск с SCP и РЕФОР в дисковод А, после чего закройте его защелку. Начнется загрузка системных программ SCP в оперативную память компьютера. По ее завершении вы увидите на экране символы **A>**.

Установите режим ввода русских букв

Нажмите клавишу **SI/SO**. При этом должна загореться индикаторная лампочка рядом с этой клавишей. Отработайте переход в новый режим нажатием любой клавиши, например пробела.

Загрузите в память РЕФОР

Введите команду: **а >** рефор **ЕТ**. В результате на экране появится соответствующая заставка, а примерно секунд через двадцать вы окажетесь в системном меню. Для более быстрого перехода в это меню нажмите клавишу пробела.

Установите уровень помощи

Введите команду **х**, а затем на запрос РЕФОРа — цифру 3 (значение, соответствующее максимальному уровню помощи).

Задайте режим создания текста

Находясь в системном меню, введите следующую команду **д**, определяющую в дальнейшем режим создания произвольного текстового документа (в отличие от создания текста программы на каком-либо алгоритмическом языке при вводе команды **н**).

Создайте файл для вашего текста

Выберите и введите имя файла (до восьми символов), под которым создаваемый текст будет сохранен на диске, например **открытка** (указание типа файла необязательно). Заметьте, что при запросе имени файла на экране высвечивается локальное меню, которое подсказывает вам, какими командами при необходимости следует воспользоваться для корректировки вводимого имени. Введя **открытка ЕТ**, вы окажетесь в основном меню РЕФОРа.

Обратите внимание!

●●● В верхней строке экрана отображается служебная информация:

◆ имя текстового файла;

◆ номера страницы, строки и позиции в строке экрана (колонки), в которых находится в данный момент курсор;
◆ подтверждение режима вставки, устанавливаемого автоматически.

●●● Служебная область отделяется от рабочей трафаретом. Эта строка содержит следующие символы:
◆ л и р указывают соответственно на левую и правую граничные позиции строки текста;
◆ ! фиксируют позиции табуляции;
◆ — отмечают остальные позиции в строке (всего таких символов первоначально 65).

●●● Крайняя правая колонка экрана в его рабочей области называется колонкой флагов. В этой колонке могут появиться следующие символы:
◆ **пробел** — данная строка содержит текст начала или внутренней части абзаца;
◆ < — помеченная строка — конец абзаца, заканчивающегося символом ЕТ;
◆ : — пустая строка на экране находится над текстом (символ может появиться при просмотре текста по направлению к его началу);
◆ . — пустая строка на экране находится под текстом (символ может появиться при просмотре текста по направлению к его концу);
◆ + — на экране отображается лишь часть строки (длина строки текста превышает длину строки экрана);
◆ П — строка — разделитель страниц текста;
◆ — — следующая строка текста будет распечатана поверх данной строки (при печати с наложением строк);
◆ ? — в строку включена неполная или нераспознанная команда управления печатью (команда с точкой);
◆ м — в строку введена команда включения данных при печати (команда слитной печати).

Установите длину строки текста

Устанавливаемая автоматически при входе в РЕФОР длина строки формируемого текста, равная 65 символам, вас не должна устроить, так как при печати поздравления на почтовой открытке могут разместиться, скажем, 50 символов. Поэтому воспользуйтесь командой **″ор** для смещения влево правой границы строки текста. В ответ на запрос РЕФОРа введите номер новой крайней правой позиции — 50. Таким образом вы установили размер строки текста длиной в 50 символов, что подтверждается выводом соответствующей строки-трафарета.

Ввод текста

Итак, позади все необходимые подготовительные операции. Теперь можно вводить непосредственно текст новогодних поздравлений. Начнем с подготовки текста поздравления, адресованного вашему коллеге:

Дорогой Константин Константинович, поздравляю Вас с Новым годом! Искренне желаю Вам крепкого здоровья и долгих лет жизни. Пусть этот год принесет Вам счастье и благополучие. Пусть в этом году исполнятся все Ваши желания. Бодрости Вам и хорошего настроения. С уважением, Иван Петров.

Как и ранее, постарайтесь следовать нашим советам.

●●● Вводите текст с первой же позиции на экране и далее непрерывно, не заботясь о переносах на следующие строки слов, выходящих за правую границу текста, и воздерживаясь от завершения строк символом **ЕТ**. В случае выхода слова за установленную границу РЕФОР автоматически обеспечит перенос его в начало следующей строки, а данную строку выровняет по правой границе за счет вставки дополнительных пробелов между словами.

●●● При вводе прописных букв, точки, запятой удерживайте в нажатом положении клавишу временной фиксации верхнего регистра клавиатуры.

●●● Если вы нажали не ту клавишу, то ошибочно введенный символ удаляйте нажатием клавиши **DEL**.

●●● Если вы с запозданием обнаружили ошибки в уже введенных строках, то не спешите прерывать ввод текста и переходить к исправлениям. Вы их исправите в дальнейшем.

В результате на экране вы увидите текст:

```
Дорогой Константин Константинович,  поздравляю Вас
с   Новым   годом!  Искренне  желаю  Вам  крепкого
здоровья  и  долгих  лет  жизни.  Пусть  этот  год
принесет Вам счастье и благополучие.  Пусть в этом
году исполнятся все Ваши желания.  Бодрости Вам  и
хорошего настроения. С уважением, Иван Петров.
```

Что вы хотите редактировать?

Допустим, при критическом рассмотрении полученного на экране текста вам кое-что в нем не понравилось. Во-первых, слово «искренне» придает поздравлению некоторую слащавость. Вы решаете удалить слово. Во-вторых, предложение «Пусть в этом году исполнятся все Ваши желания» кажется вам лишним. В-третьих, вы вспомнили, что ваш коллега Константин Константинович женат на Ирине Петровне, с которой вы, кстати, знакомы. В таком случае этикет требует поздравления обоих супругов, и вы намереваетесь начать его так: «Дорогие Ирина Петровна и Константин Константинович!». Последнее исправление повлечет за собой замену всех «Вас» и «Вам» на соответственно «вас» и «вам». Кроме того, было бы эстетичнее выделить имена с отчествами поздравляемых в отдельные строки. Желательно также выделить вашу подпись.

И со всеми этими исправлениями вам придется смириться, что в дальнейшем скомпенсируется освоением ряда эффективных средств РЕФОРа. Действия, которые позволят вам реализовать все намеченные исправления, и составляют суть процесса редактирования текста.

Редактирование текста

Выполнить все необходимые операции редактирования текста новогоднего поздравления вам поможет табл. 3.1. В первых ее двух графах приведены общепринятые в издательской деятельности корректурные знаки и собственно исправления, которые обычно изображаются на полях рукописи. В третьей графе приведен фрагмент машинописного текста на бумаге с введенным в него корректурным знаком. В четвертой графе показан тот же самый фрагмент текста на экране дисплея, где начало исправления помечено курсором (—). Порядок нажатия клавиш и необходимые при этом пояснения приведены в пятой и шестой графах. В операции замены используются

Таблица 3.1

Технология редактирования текста

Корректура		Текст		Нажимаемые клавиши	Пояснение
Корректурный знак	Исправление	на бумаге	на экране		
√	Ирина Петровна и	Дорогой√Константин	Дорогой Констан-тин	**Ирина Петровна и**	Перед вводом вставляемого текста удостоверьтесь, что установлен режим вставки
	ие	Дорогой	Дорогой	**INS ие**	Предварительно нажатием клавиши **INS** отмените режим вставки
		~~Искренне~~	Искренне	**″т**	Удалите слово справа от курсора
⊥	Ж	желаю	желаю	**Ж**	При отмененном режиме вставки замените строчную букву на прописную
		Пусть в этом ... желания. Бодрости	<б>Пусть в этом . . .	**F7**	Введите маркер начала блока, нажав клавишу **F7**. После этого на экране перед словом «Пусть» появятся символы <б>. Далее переведите курсор в начало слова «Бодрости»
			желания.<к>Бодрости	**F8**	Введите маркер конца блока, нажав клавишу

					F8. В результате после слова «желания» появятся символы <к>
				"кы	Удалите выделенный блок
[illegible] [illegible] [illegible] [illegible]	в в в в	Вас Вам Вам Вам	(Курсор может находиться в любом месте текста)	"яа В ЕТ в ЕТ нг ЕТ	Начните операцию поиска и замены прописных букв «В» на строчные «в». В служебной области экрана в ответ на запрос РЕФОРа введите следующее: ПОИСК: ЗАМЕНИТЬ НА: ПАРАМЕТР:
Z Z Z		ПетровнаZи Константинович,Z настроения.Z	Петровна_и Константинович, _ настроения. _	ЕТ ЕТ ЕТ	Выделите в отдельные строки имена с отчествами, а также вашу подпись. Для этого нажатием клавиши **INS** установите режим вставки и введите в соответствующие позиции символ ЕТ
			Дорогие и Константин С уважением,	F2 F2 F2	Переведите курсор в первую строку текста и нажатием клавиши F2 обеспечьте ее центровку. Аналогично выполните центровку второй и последней строк текста

два параметра: **н** — автоматический поиск и замена без дополнительного подтверждения пользователем; **г** — глобальный поиск и замена по всему тексту.

В результате редактирования согласно действиям, описанным в табл. 3.1, вы увидите на экране текст, в котором нарушены равномерное распределение текста по строкам и выравнивание его правой границы. Поэтому выполните форматирование текста с третьей по пятую строку поздравления с помощью трехкратного выполнения команды **″б**. В результате вы получите на экране следующий текст:

```
              Дорогие Ирина Петровна
          и Константин Константинович,
поздравляю  вас с Новым годом!  Желаю вам крепкого
здоровья   и   долгих   лет   жизни.   Пусть   этот   год
принесет вам счастье и благополучие.  Бодрости вам
и хорошего настроения.
              С уважением, Иван Петров.
```

Установите режим печати

Очередная ваша задача — отпечатать подготовленный текст на почтовой открытке. С этой целью предварительно обеспечьте режим печати прописных и строчных букв русского алфавита. Данный режим вы можете установить двумя способами. Первый из них состоит в перемещении вправо верхнего движка восьмого микропереключателя под крышкой печатающего устройства. Сделать это следует еще до его включения. Второй способ заключается в программной установке режима командой РЕФОРа **″пы**, которую введите перед первой строкой текста поздравления. Для этого предварительно зарезервируйте пустую строку перед текстом, введя в режиме вставки символ **ЕТ** в левую крайнюю позицию его первой строки.

Далее введите "**пы**, что отобразится на экране как "**ы**. Последняя операция установит для печатающего устройства режим вывода букв русского алфавита.

Рекомендуем вам также воспользоваться командами управления шрифтами печати. Так, если вы желаете отпечатать имена и отчества в поздравлении жирным шрифтом, то введите до и после имен с отчествами команду "**пб**, отображаемую на экране как "**б**. Можете также свою подпись для разнообразия отпечатать мелким шрифтом «элита», для чего непосредственно перед подписью введите "**па**, а после нее "**пн**. На экране эти команды отобразятся соответственно как "а и "н. Для подавления печати номера страницы можно ввести в конце команду **.оп ЕТ**.

```
"ы
"б                 Дорогие Ирина Петровна
               и Константин Константинович,"б
поздравляю  вас с Новым годом!  Желаю вам крепкого
здоровья  и  долгих  лет  жизни.  Пусть  этот  год
принесет вам счастье и благополучие.  Бодрости вам
и хорошего настроения.
"а                  С уважением, Иван Петров."н
```

Печатайте!

До запуска печати воспользуйтесь командой "**кс**, которая обеспечит запись хранящегося в оперативной памяти компьютера текста на магнитный диск. Дело в том, что распечатке могут быть подвергнуты только файлы, которые хранятся на магнитном диске. Включите печатающее устройство и заправьте его почтовой открыткой. Введите команду "**кп** — запуска печати текстового файла. При реализации этой команды в ответ на запрос РЕФОРа введите имя распечатываемого файла **открытка**, после

чего нажмите клавишу **ESC**. В результате вы получите распечатку новогоднего поздравления:

```
              Дорогие Ирина Петровна
          и Константин Константинович,
 поздравляю вас с Новым годом! Желаю вам крепкого
 здоровья и долгих лет жизни. Пусть этот год
 принесет вам счастье и благополучие. Бодрости вам
 и хорошего настроения.
           С уважением, Иван Петров.
```

3.3. ПЕЧАТЬ ТЕКСТОВ ПОЗДРАВЛЕНИЙ — НА КОНВЕЙЕР

Ниже вы познакомитесь еще с одним замечательным свойством РЕФОРа, которое значительно облегчит ваши предпраздничные хлопоты, связанные с подготовкой поздравительных открыток вашим друзьям и многочисленным знакомым.

Подготовьте единый текст поздравления, и РЕФОР обеспечит при печати вставку в этот текст различных имен, например: Сережа, Витя и т. д.

Повторите вышеприведенную последовательность шагов по подготовке поздравления вашему коллеге, заканчивавшуюся переходом в основное меню РЕФОРа. При этом файлу, где будет храниться текст, присвойте имя **поздрав**. Введите текст поздравления с необходимыми добавлениями:

```
"ы
.дм введите имя
.аж имя
"р      Дорогой &имя&! "е
Поздравляю тебя с наступающим Новым годом! Желаю
тебе успехов в труде, счастья в личной жизни и
всяческого благополучия!
              С приветом, Ваня Петров.
```

Ввод первых четырех строк завершите, нажав клавишу **ЕТ**. Теперь уясните назначение команд, включенных в первые четыре строки текста:

.дм обеспечивает при печати вывод на экран текста сообщения, помещенного непосредственно за командой;

.аж приостанавливает печать текста и запрашивает ввод с клавиатуры значения; оно присваивается переменной, имя которой следует непосредственно за командой;

″пы устанавливает режим печати букв русского алфавита (отображается как **″ы**);

″пр, ″пе указывают соответственно начало и конец текстового фрагмента, распечатываемого крупным шрифтом вразрядку (отображаются как **″р** и **″е**).

Подавить печать номера страницы можно, введя команду **.оп.**

Закончив ввод текста, запишите его в файл на гибком магнитном диске с помощью команды **″кд**. Находясь в системном меню, введите команду **м** — запуска так называемой слитной печати. В процессе реализации данной команды поддерживайте диалог с РЕФОРом по следующему сценарию:

Слитн. печать какого файла? поздрав ЕТ

Если ответ ″Да″ на все след. вопросы=ЕСЦ, а отд. вопросы=ЕТ

Печать в файл (д/н)? н

Старт со страницы номер (ЕТ — с начала)? ЕТ

Стоп после страницы (ЕТ — до конца)? ЕТ

Количество копий (ЕТ — одна)? 10 ESC

Вставляйте открытки в принтер и, вводя команду **П**, поддерживайте диалог по такому сценарию:

введите имя

имя? Сережа **ЕТ**

. . .

введите имя

имя? Витя **ЕТ**

. . .

В результате вы получите распечатки открыток:

```
                 Дорогой  Сережа!
Поздравляю  тебя с наступающим Новым годом!  Желаю
тебе  успехов в труде,  счастья в личной  жизни  и
всяческого благополучия!
                 С приветом, Ваня Петров.



                 Дорогой  Витя!
Поздравляю  тебя с наступающим Новым годом!  Желаю
тебе  успехов в труде,  счастья в личной  жизни  и
всяческого благополучия!
                 С приветом, Ваня Петров.
```

3.4. ОБЩАЯ СПРАВКА О РЕФОРе

Приобретя некоторые навыки работы с РЕФОРом, вы сможете расширить свои возможности при создании текстов. Для этого рекомендуем вам ознакомиться с содержанием табл. 3.2, где все команды сгруппированы по функциональному назначению. Большинство команд вводите, одновременно нажимая клавишу **CTRL** и русскую букву. Клавиша **CTRL** условно обозначена в таблице символом ″. Некоторые команды дублируются функциональной клавишей, что создает дополнительные удобства в работе.

Большинство команд, описанных в табл. 3.2, выполняются в основном меню РЕФОРа. Если же команда вводится в системном меню, то это особо отмечается в пояснении команды.

Часть команд РЕФОРа обеспечивает установку или отмену какого-либо режима работы. В табл. 3.2 режим, устанавливаемый автоматически еще до первого исполнения таких команд, выделен другим шрифтом. Выделены также значения параметров некоторых команд, устанавливаемые РЕФОРом по умолчанию.

Таблица 3.2

Команды управления РЕФОРа

Клавиша: Русский алфавит	Клавиша: функциональная	Назначение команды
		Позиционирование курсора
"с	←	Влево на один символ
"д	→	Вправо на один символ
"а	|←	Влево на одно слово
"ф	→|	Вправо на одно слово
"е	↑	Вверх на одну строку
"ь	↓	Вниз на одну строку
	←┘	Влево, в начало следующей строки
"яс		Влево, в начало данной строки
"яе		Вверх, в первую строку на экране
"яь		Вниз, в предпоследнюю строку на экране
"яд		Вправо, в конец строки
"яр		Вверх, в начало текста
"яц		Вниз, в конец текста
"я0 ... "я9		Соответственно в маркеры <0> ... <9>
"яб		В начало маркированного блока
"як		В конец маркированного блока
"яж		В позицию перед выполнением последней команды поиска или работы с блоком
"яп	F12	В позицию перед выполнением предыдущей команды
		Просмотр текста файла
"з		Вверх на одну строку
"в		Вниз на одну строку

Клавиша		Назначение команды
Русский алфавит	функциональная	
"ц	F15	Вверх на 3/4 экрана
"р	F4	Вниз на 3/4 экрана
"яз		Вверх постепенно (замедление просмотра 1 ... 9)
"яв		Вниз постепенно (замедление просмотра 1 ... 9)
		Форматирование текста
"б		Форматирование строк абзаца (до символа ЕТ)
"ох		Установка/отмена помощи при переносе слов
"ос		Установка межстрочного интервала (1 ... 9)
"ол	F1	Установка позиции левой границы текста (1 ... 240)
"ор	F3	Установка позиции правой границы текста (... 65 ... 240)
"оь		Отмена границ (до возвращения курсора в границы)
"ог		Сдвиг вправо левой границы текста до очередной позиции табуляции временно до ввода ЕТ
"ов		Установка/отмена автоматической упаковки слов при переходе к новой строке
"ой		Установка/отмена автоматического выравнивания правого края текста за счет раздвижки слов
"ое		Установка/отмена ввода так называемого мягкого дефиса для переноса слов, отображаемого на экране в виде символа подчеркивания и выводимого при печати только в случае нахождения его в конце строки
"по		Установка неразделяемости слов при переносе (в пробел между словами)
"оц	F2	Центровка строки с курсором

Клавиша		Назначение команды
Русский алфавит	функциональная	
Удаление символа, слова, строки, блока		
	DEL	Удаление символа слева от курсора
"г		Удаление символа в позиции курсора
"т		Удаление от курсора до конца слова
"яы		Удаление от курсора до конца строки
"я DEL		Удаление от начала строки до курсора
"ы		Удаление строки полностью
"кы		Удаление маркированного блока
Вставка символа, слова, строки, блока, файла		
"ж	INS	Установка/отмена режима вставки (для вставки символа, слова)
	ET	Ввод пустой строки из пробелов (для вставки строки) при нахождении курсора справа от предыдущей или в начале следующей строки (в режиме вставки)
"кж		Вставка блока с позиции курсора при удалении блока на прежнем месте
"кц	F9	Вставка блока с позиции курсора при сохранении блока на прежнем месте без маркеров (копирование блока)
"кр		Вставка файла с диска в текст
Поиск и замена фрагмента текста		
"яф	F11	Поиск запрашиваемого фрагмента (до 30 символов) с вводом параметров поиска
"яа		Поиск запрашиваемого фрагмента и его замена на вводимый текст (с подтверждением замены)
"л		Продолжение поиска того же фрагмента (после команды "яф) или поиска с заменой (после команды "яа)

Клавиша		Назначение команды
Русский алфавит	функциональная	
		Использование табуляции
"он		Отмена табуляции
"ои		Установка новых позиций табуляции (! — для обычной табуляции, # — для десятичной)
"и		Переход к следующей позиции табуляции
"оф		Автоматическая установка позиций табуляции и границ текста по строке, в начале которой находится курсор (строке-комментарию)
"ож		Установка/отмена табуляции с переменными ограничителями таблиц (для выравнивания по ограничителям)
		Управление режимами печати
"пы		Установка/отмена печати русскими буквами (прописными и строчными) и прописными латинскими буквами
"па		Установка шрифта «элита» (до 164 знаков в строке)
"пн		Восстановление стандартного шрифта «пика» (до 135 знаков в строке)
"пд		Установка/отмена двойной печати (повторной печати тех же символов)
"пб		Установка/отмена жирного шрифта (четырехкратной печати)
"пт		Установка/отмена надстрочной печати
"пж		Установка/отмена подстрочной печати
"мв		Установка печати с непрерывным подчеркиванием
"пя		Отмена печати с непрерывным подчеркиванием
"пр		Установка печати вразрядку
"пе		Отмена печати вразрядку

Клавиша		Назначение команды
Русский алфавит	функциональная	
″п ЕТ		Установка печати строки поверх предыдущей
″пх		Установка печати символа поверх предыдущего
″пц		Приостановка печати (для продолжения печати вводится команда **″кп** или **п** из системного меню)
		Управление вставками данных при печати
.аж		Запрос ввода значения переменной с клавиатуры (имя переменной указывается непосредственно за командой)
.сж		Присвоение значения переменной (за командой следуют имя и значение переменной через запятую)
.дф		Спецификация файла для чтения из него значений переменных, имена которых указываются в последующей команде **.рж** (за командой следует имя файла данных)
.рж		Чтение значений переменных из файла, предварительно определенного командой **.дф** (имена переменных следуют за командой через запятую)
.фи		Вставка в место введения данной команды текстового файла, имя которого указывается вслед за командой
.дм		Вывод на экран следующего за данной командой текста (используется для подсказки ввода командой **.аж**)
.цс		Очистка экрана
		Формирование и нумерация страниц для печати
.мт#		Установка числа пустых строк (#) от верхнего края листа бумаги до начала текста (по умолчанию # = 3)

Клавиша		Назначение команды
Русский алфавит	функциональная	
.пл#		Установка числа строк (#) на странице (по умолчанию # = 66), включая отступы сверху и снизу
.мб#		Установка числа пустых строк (#) ниже текста до конца листа (по умолчанию # = 8)
.оп		Отмена печати номеров страниц стандартного для РЕФОРа расположения (внизу — посередине)
.пн		Восстановление печати номеров страниц стандартного расположения (внизу — посередине)
.хе		Установка/**отмена** печати общего заголовка для последующих страниц (текст заголовка вводится при установке непосредственно за командой, используется при печати номеров страниц сверху)
.фо		Установка/**отмена** печати общей подстраничной надписи для последующих страниц (текст надписи вводится непосредственно за командой)
.пн		Установка печати возрастающих номеров страниц начиная со страницы, где введена данная команда
.пн#		Установка печати возрастающих номеров страниц, начиная с номера #
.па		Переход на новую страницу (вводится в конец страницы, после которой осуществляется переход)
.цп#		Переход на новую страницу, если до конца поля текста на данной странице остается менее чем # строк (используется для предупреждения возможного разрыва головок таблиц)
..		Ввод комментария, игнорируемого при печати (используется в качестве образца строки табуляции для команды "**оф**)
"п#		Вставка в заголовок или подстраничную надпись символа #, указывающего позицию для печати номеров страниц (используется в командах **.хе** и **.фо**)

Клавиша		Назначение команды
Русский алфавит	функцио-нальная	
"пк		Установка печати заголовка или подстраничной надписи слева для четных номеров страниц и справа для нечетных номеров
		Управление процессом печати
"кп		Запуск/приостанов печати файла с возможностью одновременного редактирования (имя файла запрашивается)
п		Запуск/приостанов печати файла из системного меню
м		Печать из системного меню с включением данных из другого файла или вводимых с клавиатуры (включение производится в заранее зарезервированные места текста)
н		Продолжение печати, приостановленной командами "кп и п системного меню
д		Прекращение печати
		Работа с файлами
д		Открытие файла текстового документа (из системного меню)
н		Открытие файла исходной программы (из системного меню)
"ко		Копирование файла
о		То же из системного меню
"ке		Переименование файла
е		То же из системного меню
"кй		Удаление файла
ы		То же из системного меню
"кл		Переназначение текущего дисковода

Клавиша		Назначение команды
Русский алфавит	функциональная	
л		Переназначение текущего дисковода из системного меню
″кф		Установка/отмена вывода на экран каталога диска
″кв		Запись маркированного блока на диск
″кр		Вставка считываемого с диска файла в место, указанное курсором
″кс	F13	Запись редактируемого файла на диск и продолжение работы в основном меню (рекомендуется вводить периодически)
″кд	F14	Запись редактируемого файла на диск и переход в системное меню
″кь		Запись редактируемого файла на диск и выход в систему SCP
ь		Выход в систему SCP (из системного меню)
″кя		Переход в системное меню без сохранения редактируемого файла
		Прочие операции
рефор		Начальная загрузка РЕФОРа (в системе SCP)
л		Переход к другому дисководу (в системном меню)
ф		Установка/отмена вывода на экран каталога диска (в системном меню)
х		Установка уровня помощи в системном меню (0— подавление вывода справочной информации; 1 — выводится только системное меню; 2 — выводятся все меню, кроме основного; 3 — выводятся все меню)
″йх		То же в основном меню
р		Запуск системной программы из системного меню (например, STAT, INIT)
ь		Выход в систему SCP (из системного меню)

Клавиша		Назначение команды
Русский алфавит	функциональная	
й		Вызов меню помощи (из системного меню)
"у		Прерывание выполняемой команды
"яя		Автоповтор вводимой вслед за "яя команды
"к0..."к9		Ввод/удаление маркера (для последующего использования команд "я0 ... "я9)
"кн	F7	Ввод/удаление маркера начала блока (<б>)
"кк	F8	Ввод/удаление маркера конца блока (<к>)
"кх	F6	Установка/отмена индикации маркеров начала и конца блока
	F10	Прогон бумаги на принтере

3.5. ПРОВЕРЯЕМ СЕБЯ. КАК УСВОЕН РЕФОР?

Упражнения

3.1. Обеспечьте вывод на экран каталога магнитного диска в основном меню РЕФОРа.

3.2. Каким образом можно удалить лишний символ в слове?

3.3. Можно ли произвести поиск образца текста в файле по направлению к его началу?

3.4. Как наиболее простым образом расположить короткую строку посередине текстового поля?

3.5. Как предотвратить возможное при форматировании текста разделение по строкам фамилии и инициалов?

3.6. Как предупредить разрыв головки таблицы из четырех строк при автоматическом разбиении текста на страницы?

3.7. Каковы ваши действия в случае, когда русский текст распечатывается в латинском алфавите?

3.8. Выполните печать подготовленного с помощью РЕФОРа документа в нескольких экземплярах.

Ответы к упражнениям

3.1. Введите команду ″**кф**, в результате выполнения которой в служебной области экрана появится каталог магнитного диска. С помощью той же команды вы можете подавить вывод каталога.

3.2. Предварительно переместите курсор в позицию удаляемого символа. Затем введите команду ″**г**.

3.3. При использовании команд поиска ″**яф** и поиска с заменой ″**яа** вы можете задать направление поиска к началу текстового файла, если в список параметров поиска введете символ **б**.

3.4. Для центровки короткой строки переведите в нее курсор, а затем нажмите клавишу **F2**.

3.5. Для предотвращения переноса инициалов на следующую строку введите в пробелы после фамилии и между инициалами команду ″**по**.

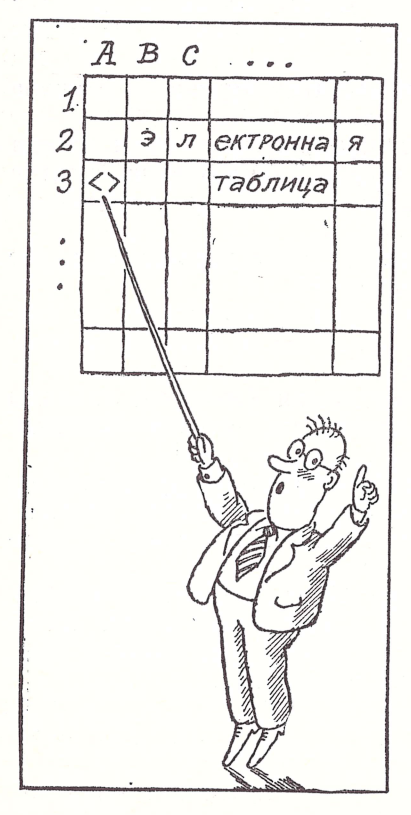
3.6. Введите перед текстом шапки (головки) таблицы в предварительно зарезервированную строку команду **.цп4**.

3.7. Выключите печатающее устройство, приподнимите его крышку и поменяйте положение верхнего движка восьмого микропереключателя. Затем включите устройство и повторите вывод на печать.

3.8. Завершите подготовку текста документа нажатием клавиши **F14** с переходом в системное меню. Далее воспользуйтесь командой **м**, введя при подготовке к печати желаемое количество копий документа.

4

ПРИМЕНЯЕМ ВАРИТАБ — ЭЛЕКТРОННУЮ ТАБЛИЦУ



4.1. ЗАЧЕМ НУЖНА ЭЛЕКТРОННАЯ ТАБЛИЦА?

Сфера использования

В процессе производственной деятельности любого специалиста часто требуется представить результаты работы в виде таблиц, где одна часть граф занята исходными данными, а другая — результатами вычислений и анализа. Такая форма работы благодаря своей наглядности и простоте настолько широко распространена, что охватывает практически любую сферу деятельности. Даже школьники, проводя эксперименты, строя графики функций, исходные данные и результаты экспериментов фиксируют в таблице. На основании этих данных в

дальнейшем производят расчеты и заполняют остальные графы.

Используют таблицы особенно широко в административной, экономической, хозяйственной сферах. Расчеты, результаты которых вносят в таблицы, как правило, очень просты и для их проведения не требуется специальной математической подготовки. Характерными для них являются большие объемы перерабатываемой информации. Часто возникает необходимость в многократных просчетах, в ходе которых умышленно вносят некоторые изменения в исходные данные, т. е. нужно провести численное моделирование той или иной ситуации.

Расчеты при больших массивах данных — задача утомительная, требующая повышенного внимания, и в некотором смысле неблагодарная. Велик процент появления ошибок. Все это позволяет отнести такую работу к разряду рутинных.

Автоматизация расчетов с помощью электронной таблицы

Необходимо средство, которое позволит автоматизировать процесс проектирования таблиц. В качестве такого средства служат специализированные программы, ориентированные на расчеты по одной и той же табличной форме. Недостаток этих программ — отсутствие универсальности. Другим, более универсальным средством являются пакеты прикладных программ (ППП), используя которые можно проектировать различные формы таблиц и производить вычисления с помощью простого входного языка.

К числу таких ППП относится пакет прикладных программ для работы с электронной таблицей. Электронной называется таблица, создаваемая в компьютере посредством организации диалога между пользователем и ЭВМ. Пакет прикладных программ для работы с электронной таблицей часто называют табличным процессором или ва-

риационной (вариантной) таблицей. Семейство подобных пакетов, созданное первоначально с целью облегчить расчеты по табличной форме, нашло более широкую сферу применения — моделирование различных ситуаций. В считанные минуты вы можете получить ответ на вопрос «Что будет, если ...?» из таблицы, содержащей несколько сотен расчетных данных.

Суть автоматизации расчетов с помощью электронной таблицы заключается в следующем:

◆ ускоряется и облегчается процесс формирования таблицы (заголовка и шапки) и заполнения ее данными;

◆ внесение изменений хотя бы в одну клетку таблицы вызывает автоматический пересчет всей таблицы;

◆ если в таблице часто используется одна и та же формула расчета для разных переменных, то имеется возможность не писать ее многократно в каждую клетку, а автоматизировать этот процесс с помощью одной команды;

◆ автоматизирован процесс печати таблицы;

◆ просты для освоения и в эксплуатации средства редактирования таблицы;

◆ для персональных компьютеров с 16-разрядным микропроцессором можно формировать новые таблицы, производя выборку из существующих, аналогично тому, как делается это при работе с базами данных, а также оформлять данные в виде графиков.

Обзор

Одним из наиболее известных табличных процессоров является пакет прикладных программ SUPERCALC (Суперкалк), имеющий несколько версий для разного класса персональных компьютеров. В отечественной практике широкое распространение для персонального компьютера Роботрон 1715 получила адаптированная на русский язык версия SUPERCALC под названием ВАРИТАБ (ВАРиантная ТАБлица). Этот пакет находится на сопровождении в НПО «Центрпрограммсистем».

В другой версии, например SUPERCALC-2 (C2), также адаптированной на русский язык, увеличен объем таблицы (255×127), добавлено несколько функций, в том числе сортировки, сделан вывод информации в файлы, совместимые с языком BASIC и редактором текста WORDSTAR. В некоторых версиях введено более удобное средство перемещения курсора с помощью клавиш направления вместо управляющих символов, где требуется одновременное нажатие двух клавиш. Принципы построения электронной таблицы заложены и в пакет прикладных программ MULTIPLAN (MP), предназначенный для плановых и статистических расчетов.

Применительно к персональным компьютерам с 16-разрядным микропроцессором электронная таблица представляет собой не только инструмент для обработки и формирования таблиц, но и мощный интегрированный пакет прикладных программ, где дополнительно используются:

◆ графические средства, позволяющие оформлять данные в виде различных диаграмм — столбиковой, круговой, линейной и т. д.;

◆ средства организации баз данных, где появляется возможность выборки данных в соответствии с критерием.

К таким пакетам, например, относятся LOTUS-1-2-3, ВАРИТАБ-86.

Характеристика ВАРИТАБа

Представленный ниже пакет прикладных программ ВАРИТАБ отражает основные принципы построения электронной таблицы, ознакомившись с которыми вы без труда будете работать с любой версией электронной таблицы. ВАРИТАБ занимает на диске объем 64 Кбайт. Работа по формированию таблицы, вводу исходных данных, расчетам, выводу табличной формы осуществляется с помощью 19 команд в диалоговом режиме. Вам предлагается вопрос и приводится перечень ответов, на одном из которых вы останавливаетесь. Максимальный объем

электронной таблицы составляет 254 строки и 63 столбца. В электронной таблице строки в дальнейшем будут называться рядами. Нумерация рядов идет от 1 до 254. Столбцы электронной таблицы называются колонками. Каждая колонка имеет имя (номер) в виде одной или двух латинских букв: A, B, C, D, ..., Z, AA, AB, AC, ..., BJ, BK.

Прежде чем работать с диском, уточните, записана ли на нем операционная система. В противном случае вам необходимо записать ее на первые три дорожки диска, отводимые всегда для операционной системы.

Как быстро изучить ВАРИТАБ?

Освоить ВАРИТАБ достаточно просто, если вы последовательно ознакомитесь с содержанием этой главы, где на конкретном примере показана основная концепция работы с электронной таблицей. Работа с таблицей организуется с помощью вводимых вами команд. В качестве сквозного примера рассмотрена методика создания ведомости начисления заработной платы, организации ее расчетов и вывода результатов. Когда вы поймете механизм построения электронной таблицы и работы с ней, обратитесь к описанию команд, их основных функций и структуры и на конкретном примере проследите поэтапное выполнение каждой команды.

Обратите внимание!

●●● Работайте с электронной таблицей, используя регистр русского алфавита. Для этого нажмите клавишу **ФВР** — фиксации верхнего регистра. Загорание лампочки индикации, находящейся в левой части клавиатуры, сообщает вам о фиксации русского алфавита.

●●● В процессе работы не надо переключать регистры. В том случае, когда требуется ввести латинскую букву, достаточно просто нажать клавишу с этой буквой. Несмотря на то что вы не переключали регистр на латинский алфавит, ваши действия ВАРИТАБ воспримет правильно.

●●● Вам придется постоянно следить за правильностью ввода букв, так как некоторые буквы обоих алфавитов сходны по начертанию.

4.2. ИСПОЛЬЗУЕМ ВАРИТАБ ДЛЯ РЕШЕНИЯ КОНКРЕТНОЙ ЗАДАЧИ

4.2.1. СТАВИМ ЗАДАЧУ — РАССЧИТАТЬ ЗАРАБОТНУЮ ПЛАТУ

Для приобретения первых навыков работы с электронной таблицей предлагаем вам последовательно пройти все этапы. Начнем с постановки задачи.

Вам необходимо создать электронную таблицу, по которой ежемесячно будет производиться расчет заработной платы рабочих. Для более наглядного представления функции электронной таблицы расчет носит несколько условный, упрощенный характер и в нем учтены только основные удержания.

Исходными данными для расчета являются: фамилия, имя, отчество рабочего; тариф T; количество отработан-

Таблица 4.1

Подоходный налог N (при $Z \leqslant 100$ руб.)

Заработная плата Z, руб.	Налог N, руб. и коп.
...	...
95	7.60
96	7.72
97	7.84
98	7.96
99	8.08
100	8.20

Таблица 4.2

Дополнительная часть N_1 подоходного налога (при $Z > 100$ руб.)

Заработная плата Z, руб.	Налог N_1, руб. и коп.
$100 < Z \leqslant 200$	1.30
$200 < Z \leqslant 300$	2.60
$300 < Z \leqslant 400$	3.90
$400 < Z \leqslant 500$	5.20
$500 < Z \leqslant 600$	6.50

ных дней K. Надо рассчитать заработную плату Z, подоходный налог N, профсоюзные взносы W по формулам:

$$Z = TK; \ W = 0{,}01Z.$$

Подоходный налог зависит от суммы заработной платы:

◆ если $Z \leqslant 100$ руб., то для определения подоходного налога N следует воспользоваться табл. 4.1, где Z — заработная плата в диапазоне 70—100 руб.;

◆ если $Z > 100$ руб., то $N = 8.20 + N_1$, где N_1 определяется из табл. 4.2.

Сумма к выдаче на руки рабочему рассчитывается по формуле:

$$S = Z - W - N.$$

Необходимо определить также итоги по графам «Заработная плата» и «Сумма к выдаче» последовательным суммированием каждого нового полученного данного к предыдущей сумме. Все исходные данные и результаты

Таблица 4.3

Ведомость начисления заработной платы

Фамилия, имя, отчество	Тариф	Кол. отраб. дней	Заработная плата	Удержание		Сумма к выдаче
				Проф. взносы	Подох. налог	
Корнев К. К. Лапин Л. Л. Маркова М. М.	7.20 8.90 5.50	20 25 18	?	?	?	?
Итого			?			?

расчетов должны быть оформлены в виде табл. 4.3. Для общей наглядности предлагаем провести расчет заработной платы для трех рабочих.

4.2.2. ЗНАКОМИМСЯ СО СТРУКТУРОЙ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

Начало работы

На диске должна находиться операционная система (первые три дорожки) и ВАРИТАБ. После установки диска в дисковод А вызовите каталог и найдите имя основной программы пакета. В одной из версий это имя WT. Введите имя программы

A>WT

На экране появится таблица, состоящая из восьми граф A, B, C, D, E, F, G, H и 20 пронумерованных рядов — 1, 2, ..., 20 (рис. 4.1).

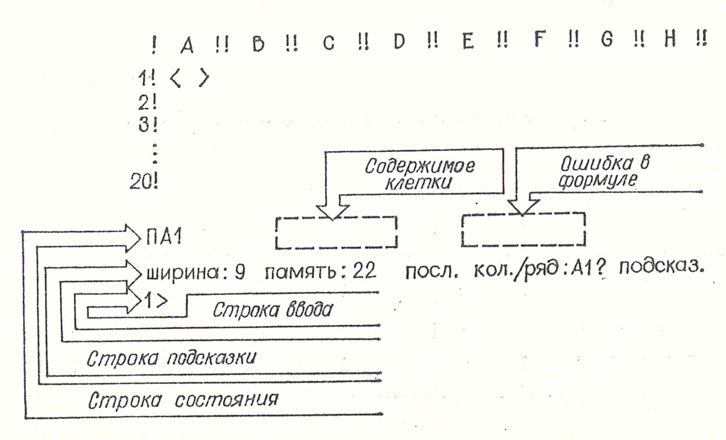


Рис. 4.1. Структура электронной таблицы

Указатель и текущая таблица

На рис. 4.1 вы видите в клетке А1 два символа < >. Это указатель клетки, куда будет введено данное. В дальнейшем клетку, ряд, графу, где установлен указатель, будем называть текущими. Кроме того, текущей является таблица, находящаяся в оперативной памяти и отображенная на экране дисплея.

Указатель по таблице можно передвигать с помощью управляющих символов. Для этого следует одновременно нажать две кавиши:

CTRL S или ← — влево; **CTRL E** — вверх;
CTRL D — вправо; **CTRL X** — вниз.

В некоторых версиях в электронной таблице для передвижения указателя используются клавиши направления:

← — влево; ↑ — вверх;
→ — вправо; ↓ — вниз.

Для отмены (сброса) набранной команды или данного будете нажимать одновременно клавиши **CTRL Z** или в иной версии |← .

Под таблицей расположены три строки, которые служат вам для оказания помощи и ведения диалога.

Строка состояния

В этой строке отображаются три типа сообщений.

●●● Первое сообщение включает название текущей клетки и направление движения указателя. Например, на рис. 4.1 ПА1 означает, что указатель находится в клетке А1 и при дальнейшей работе будет перемещаться вправо на одну клетку (буква П). Возможные варианты движения вы определите благодаря их обозначению: П — вправо; Л — влево; В — вверх; Н — вниз.

●●● Второе сообщение выводится в середине строки и показывает содержимое текущей клетки. Если клетка

пустая, то это сообщение отсутствует, что и отображено на рис. 4.1. Для вывода типа данного, находящегося в клетке, используются следующие условные обозначения: для формул и чисел — ФОРМ, например: ФОРМ= =2+3*7.2; для текста — ТКСТ, например: ТКСТ="ПЛАН; для повторяющегося в нескольких клетках текста — ПТКСТ, например: ПТКСТ='*.

●●● Третье сообщение появляется довольно редко в конце строки состояния, так как оно служит для указания ошибки при вводе данных и команд.

Строка подсказки

Строка подсказки имеет двойное назначение.

●●● С помощью этой строки организуется диалог между пользователем и электронной таблицей в процессе выполнения команды. Здесь формируется сообщение, которое подскажет вам, какую клавишу нажать для ввода параметра команды. Кроме того, в любой момент вы, не зная, как поступить дальше, можете попросить у ВАРИТАБа помощи, нажав одновременно клавишу временного перехода на верхний регистр (**ВПДР** на рис. 1.5) и клавишу **?**. В других версиях то же самое можно получить, нажав функциональную клавишу **F1**. В этом случае на экране появится разъяснение интересующего вас действия. Например, вы нажали клавишу **/**, и в строке подсказки появится сообщение:

Укажи А, Б, В, Г, Д, З, И, К, Л, М, Н, О, П,
Р, С, Т, У, Ф, Я?

Вы еще не знаете, что означают эти буквы. Нажмите одновременно клавиши **?** и **ВПДР**, на экране появится перечень команд ВАРИТАБ. Ознакомьтесь с ним. Для возврата к исходному состоянию нажмите клавишу **ЕТ**. На экране вновь появится исходная таблица с предложением ввести одну из букв. Теперь вы уже знаете, что ввод каждой буквы равносилен вводу команды. Не бойтесь нажимать клавишу подсказки **?**. Это поможет вам

лучше освоить методику работы с электронной таблицей.

●●● Вторым назначением строки подсказки является предоставление вам сведений о характеристиках текущей таблицы.

◆ ШИРИНА означает ширину текущей клетки, где установлен указатель. По умолчанию, если вы не проводили форматизации клетки, ее ширина равна девяти позициям. Дополнительная позиция для разделения клеток не отводится. Например, на рис. 4.1 ширина всех колонок, а значит, и ширина клеток равна девяти позициям;

◆ ПАМЯТЬ указывает объем оперативной памяти в килобайтах для текущей таблицы. Например, на рис. 4.1 для текущей таблицы выделено 22 Кбайт оперативной памяти.

◆ ПОСЛ.КОЛ/РЯД означает последнюю занятую клетку, находящуюся справа внизу текущей таблицы. Это равносильно указанию ее размеров, т. е. последней колонки и последнего ряда. Данное сообщение особенно важно при размещении таблицы на нескольких экранах. Вы всегда будете знать, какой колонкой и каким рядом она ограничена. На рис. 4.1 выведено сообщение «ПОСЛ. КОЛ/РЯД:A1». Вы не заполняли таблицу, поэтому первая клетка является и последней.

Например, если сообщение имеет вид

ПОСЛ.КОЛ/РЯД: E16,

то таблица содержит пять колонок (A, B, C, D, E) и 16 рядов.

Строка ввода

Здесь отображается информация, вводимая с клавиатуры. Текущая позиция, куда вводится очередной символ, указана курсором (символ —), а цифра, находящаяся слева от символа > (см. рис. 4.1), означает номер позиции курсора.

●●● Для ввода данных в текущую клетку надо нажать клавишу **ЕТ**. При этом строка ввода очистится.

●●● При вводе команд строка ввода сохраняется до тех пор, пока она не будет полностью сформирована. Ввод промежуточных ответов на каждую подсказку при формировании команды лишь дополняет строку ввода.

4.2.3. ФОРМИРУЕМ ЗАГОЛОВОК И ШАПКУ ТАБЛИЦЫ

После знакомства с исходной таблицей, изображенной на рис. 4.1, вы можете приступить к формированию таблицы для расчета заработной платы. Сначала сформируйте заголовок и шапку (головку) таблицы, а затем приступайте к вводу данных.

Для изучения основных средств языка предлагаем вам ограничиться вначале первыми четырьмя графами табл. 4.3. К формированию остальных граф вы приступите, освоив эти минимальные средства и желая более четко уяснить возможности использования электронной таблицы (см. параграф 4.3).

Ввод заголовка

Ввод любого текста всегда начинается с кавычек ″, а затем на клавиатуре набираются любые символы.

Обычно заголовок располагают симметрично относительно центра. Проведите предварительную работу по расчету ширины табл. 4.3. В каждой графе определите максимальное количество символов в данном, добавьте для разделения каждой графы символ «пробел», и вы получите ширину документа. Ширина первой графы равна 20 позициям для записи фамилии и инициалов, второй графы — пяти позициям для записи слова «Тариф». Ширина третьей графы задается словом «Отраб.» и равна шести позициям, а четвертой графы — словом «Заработ-

ная» и равна 10 позициям. Для отделения граф дополнительно используйте три позиции. Всего ширина документа равна 20+5+6+10+3=44 позициям. Середина документа — 22-я позиция.

Заголовок введите двумя строками ввода, так как он состоит из двух рядов (см. табл. 4.3). Установите указатель в клетке A1. Для формирования текста в первом ряду нажмите клавишу ″, затем 16 раз — клавишу пробела, наберите слово ВЕДОМОСТЬ. Тогда строка ввода примет вид

```
27>"                ВЕДОМОСТЬ
```

Для ввода этой строки в таблицу нажмите клавишу **ET**. Для формирования текста во втором ряду установите указатель в клетке A2, нажмите клавишу ″, затем шесть раз — клавишу пробела, наберите слова НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ.

Строка ввода примет вид

```
35>"      НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ
```

Для ввода этой строки в таблицу нажмите клавишу **ET**. Чтобы отделить заголовок от таблицы, пропустите пустой ряд, переведя указатель в клетку A4. Так вы закончили формирование заголовка электронной таблицы, который разместился в четырех колонках A, B, C, D.

```
  !    A    !!    B    !!    C    !!    D
 1!                ВЕДОМОСТЬ
 2!      НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ
 3!
```

Далее приступайте к формированию шапки таблицы, последовательно устанавливая ширину каждой колонки и вводя название, а также выделяя границы таблицы горизонтальной чертой.

Ввод горизонтальной черты

Вы начинаете формировать горизонтальную черту, изображая ее многократно повторяющими символами = или —. Для организации многократного повторения в электронной таблице используется апостроф ′. Установите указатель в клетке А4, нажмите клавиши ′ и =. В строке ввода появится

3>′=

Нажав клавишу **ЕТ**, вы увидите, что символ = будет введен во все позиции клеток ряда 4.

Установка ширины колонки А и ввод названия

●●● Установите указатель в клетке А5. Вам надо изменить ширину графы, сделав ее равной 20 позициям. Воспользуйтесь командой изменения форматов F(FORMAT). При вводе любой команды необходимо нажать две клавиши: первой будет символ / (слэж), а второй — первая латинская буква в названии команды. Вы еще не знакомы с командами?

Помните! У вас в руках мощное средство подсказки. Воспользуйтесь им. Нажмите клавишу /, затем одновременно клавишу подсказки **?** и **ВПДР**. На экране появится перечень команд ВАРИТАБа (см. табл. 4.6). Ознакомьтесь с ним, а затем, нажав клавишу ввода **ЕТ**, вы будете возвращены к исходному состоянию экрана.

После нажатия клавиши / состояния строк подсказки и ввода изменятся:

строка подсказки

ВВЕДИТЕ БУКВУ А, Б, В, Г, Д, З, И, К, Л, М, Н, О, П, Р, С, Т, У, Ф, Я?

строка ввода

2>/

Для ввода команды нажмите клавишу **Ф** (**F**). Строка

подсказки приглашает **вас ответить на** вопрос, а в строке ввода появится слово FORMAT:

строка подсказки

УКАЖИ ОБЛАСТЬ: ТАБЛИЦА, КОЛОНКА, RЯД ИЛИ ЭЛЕМЕНТ

строка ввода

9>/FORMAT,

Вам надо увеличить ширину колонки, поэтому следует нажать клавишу **К**. Вновь изменятся состояния строк подсказки и ввода:

строка подсказки

УКАЖИ БУКВУ КОЛОНКИ

строка ввода

16>/**FORMAT, КОЛОН.,**

Укажите букву колонки, нажав клавишу **А** и клавишу **ЕТ**. В строке подсказки вам будет предложено ввести форматы:

строка подсказки

ЗАДАЙ ФОРМАТЫ (C, A, N, ¤ , P, L, TP, TL, *, I. ШИР. КОЛОНКИ)

строка ввода

18>/**FORMAT, КОЛОН., А,**

Однако вы еще не знакомы с видами форматов. Воспользуйтесь клавишей помощи ?. Нажмите ее и **ВПДР**, и на экране будет изображен перечень возможных форматов. После ознакомления с содержанием экрана вернитесь к исходному состоянию, нажав клавишу **ЕТ**. Выберите требуемую ширину колонки (шир. колонки), а именно: наберите число **20** и нажмите клавишу **ЕТ**. Вы увидите, что строки подсказки и ввода очистились, а ширина графы А увеличилась до 20 позиций.

●●● Теперь займитесь вводом названия колонки. Проверьте, установлен ли указатель в клетке А5. Введите строку

15>"   ФАМИЛИЯ

Установите указатель в клетку А6 и введите строку
17>″ ИМЯ, ОТЧЕСТВО

Установка ширины колонки В и ввод названия

●●● Согласно расчету ширина колонки В равна шести позициям (5+1 — для обозначения вертикальной черты символом:). Попробуйте установить ширину графы, не пользуясь подсказкой. Переведите указатель в клетку В5 и воспользуйтесь командой F (FORMAT), так же как при формировании ширины колонки А. Наберите полностью строку ввода, нажимая клавиши в следующей последовательности: **/F К В ЕТ 6 ЕТ** (латинский алфавит) или **/Ф К Б ЕТ 6 ЕТ** (русский алфавит). После ввода каждого параметра команды на экране в строке ввода появляется условное название выбранного параметра, заканчивающееся запятой. Одновременно изменится сообщение в строке подсказки. В результате действия введенной команды произойдет изменение ширины колонки В, она станет равной шести позициям.

●●● Введите название колонки тремя строками, устанавливая указатель сначала в клетке В5, затем для ввода следующих строк — в клетках В6 и В7:

8>″:**ТАРИФ**
3>″:
3>″:

Установка ширины колонки С и ввод названия

●●● В соответствии с расчетом ширина колонки равна семи позициям (6+1 — для символа :). Для вас уже не составит труда изменить ширину этой колонки. Напомним, что надо нажать клавиши **/ F К С ЕТ 7 ЕТ** (латинский алфавит) или **/ Ф К Ц ЕТ 6 ЕТ** (русский алфа-

вит). После этого ширина колонки изменится и станет равной 7.

●●● Установите указатель в клетке C5 и введите строку

8>": **КОЛ.**

Установите указатель в клетке C6 и введите строку

9>":**ОТРАБ.**

Установите указатель в клетке C7 и введите строку

8>": **ДНЕЙ**

Установка ширины колонки D и ввод названия

●●● Согласно расчету ширина графы равна 11 позициям (10+1 — для символа :). Введите команду **/FK D ET 11 ET** (латинский алфавит) или **/ Ф К Д ЕТ 11 ЕТ** (русский алфавит). Ширина колонки D станет равной 11.

●●● Установите указатель в клетку D5 и введите строку

13>":**ЗАРАБОТНАЯ**

Установите указатель в клетку D6 и введите строку

11>": **ПЛАТА**

Установите указатель в клетку D7 и введите строку

3>":

Ввод горизонтальной черты

Заканчивается формирование шапки таблицы горизонтальной чертой. Установите указатель в клетку A8 и наберите в строке ввода повторяющийся текст

3>'=

После нажатия клавиши ввода на экране появится во всех клетках восьмого ряда символ =.

Итак, вы проделали все операции по формированию заголовка и шапки таблицы. Для проверки правильности выполнения изложенных в этом параграфе действий сверьтесь с приведенным ниже изображением заголовка и шапки электронной таблицы.

```
    !          A         !!  B !!  C  !!    D     !
  1!                ВЕДОМОСТЬ
  2!      НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ
  3!
  4!===============================================
  5!        ФАМИЛИЯ        :ТАРИФ: КОЛ.  :ЗАРАБОТНАЯ
  6!   ИМЯ,ОТЧЕСТВО        :      :ОТРАБ.:  ПЛАТА
  7!                       :      : ДНЕЙ :
  8!===============================================
  9!КОРНЕВ К.К.             7.2      20      144.00
 10!ЛАПИН Л.Л.              8.9      25      222.50
 11!МАРКОВА М.М.            5.5      18       99.00
 12!===============================================
 13!  ИТОГО                                  465.50
```

Надеемся, что все совпало. Если нет, то наберитесь терпения и проделайте еще раз те действия, которые, как оказалось, не увенчались успехом. Теперь наверняка все в порядке. Желаем дальнейших успехов и перейдем к описанию процедуры ввода данных.

4.2.4. ВВОДИМ ДАННЫЕ В ТАБЛИЦУ

Ввод текста в колонку А

В колонку А внесите фамилии и инициалы рабочих. Установите указатель в клетку А9 и наберите фамилию как текстовое данное, т. е. начиная с ":

13>"**КОРНЕВ К. К.**

Введите эту строку, нажав клавишу **ET**.

Обратите внимание! Введенный текст всегда будет располагаться начиная с левого края.

Переместите указатель в клетку A10 и введите строку

12>"ЛАПИН Л. Л.

Переместите указатель в клетку A11 и введите строку

14>"МАРКОВА М. М.

Ввод чисел в колонку B

В колонке B содержатся числа, соответствующие тарифным ставкам рабочих. Установите указатель в клетку B9 и введите первое число

5>7.20

Установите указатель в клетку B10 и введите строку

5>8.90

Установите указатель в клетку B11 и введите строку

5>5.50

Здесь стоит остановиться и сделать ряд замечаний по вводу данных:

- при вводе в клетку число располагается, прижимаясь к правому краю, а текст — к левому краю;
- при вводе числа в клетку автоматически выделяется пустая позиция слева, чтобы число не сливалось с содержимым левой клетки.

Например, если ширина колонки — семь позиций, а вы вводите число —12.3567, состоящее из восьми позиций, то оно округляется и будет состоять из шести позиций, т. е. —12.36;

- при вводе текста позиция между колонками не выделяется.

Ввод чисел в колонку C

В графе C содержатся сведения о количестве отработанных дней каждым рабочим за истекший месяц. Установите указатель в клетку C9 и введите строку

3>20

Установите указатель в клетку C10 и введите строку

3>25

Установите указатель в клетку C11 и введите строку

3>18

Все введенные исходные данные изображены в электронной таблице, представленной на стр. 133.

Ввод формул в колонку D

В колонке D будут размещены результаты расчета заработной платы по формуле $Z=TK$.

Что такое формулы?

Формулой в электронной таблице называется арифметическое выражение. Вводите формулы в каждую клетку так же, как и числа. Прежде чем приступить к вводу формул, ознакомьтесь с некоторыми общими правилами их образования:

◆ формула, так же как и любое арифметическое выражение, образуется из знаков операций, круглых скобок и операндов.

Операндами могут быть числовые константы, числовые переменные и функции;

◆ соблюдается традиционный порядок выполнения действий: возведение в степень (∧); умножение (*) или деление (/), сложение (+) или вычитание (—);

◆ круглые скобки ( ) служат для изменения принятого порядка выполнения арифметических действий;

◆ в качестве числовых переменных используются названия клеток, называемые ссылками;

◆ в формулах часто оперируют наиболее употребительными в математике функциями, для которых введены условные обозначения, например: для вычисления функций e^x — ЕКС (X), для определения целой части числа — ЦЕЛ (X) и т. д.; в качестве аргументов можно использовать константу, арифметическое выражение, ссылки.

Как вводить формулы?

Приступим к вводу формул в клетки D9, D10, D11. В качестве операндов в формулах служат названия клеток (ссылки). Для вычисления заработной платы, скажем, Корневу К. К. (см. стр. 133) надо перемножить тариф (клетка B9) и количество отработанных дней (клетка C9), т. е. B9*C9. Для удобства ввода формул со ссылками переключите регистр на латинский алфавит, нажав клавишу **ФВР** (индикатор погаснет).

Установите указатель в клетке D9. В строке ввода наберите и введите

6>**B9*C9**

В клетке D9 появится число 144=7.2*20. Указатель передвинется в следующую клетку указанного направления. Убедитесь в том, что ваши действия правильны. Передвиньте указатель снова в клетку D9. Тогда в строке состояния, там, где помещается содержимое клетки, вы увидите сообщение ФОРМ=B9*C9. Это означает, что в клетку D9 была введена формула.

Установите указатель в клетку D10. В строке ввода наберите и введите

6>**B10*C10**

В клетке D10 появится число 222.5. Установите указатель в клетку D11. В строке ввода наберите и введите

6>**B11*C11**

В клетке D11 появится число 99. Располагаются числа в таблице после ввода формул, прижимаясь к правому краю, т. е.

144
222.5
99

Как форматировать числа в колонке?

Как вы понимаете, такое представление чисел в колонке D таблицы нежелательно. Надо, чтобы каждый разряд числа занимал определенное место. Для наглядного

отображения заработной платы в этой колонке следует представить числа с точностью до двух десятичных знаков. Сделать это достаточно просто, если вы воспользуетесь командой форматирования F с параметром ¤, который дает возможность представлять числа в требуемом формате. С командой F вы ознакомились раньше, при форматировании ширины колонки. Теперь воспользуйтесь ею для более наглядного изображения числа. Для ввода этой команды нажмите следующую последовательность клавиш: **/F K D ET ¤ ET** (латинский алфавит), где **/F** — название команды; **K** — указание на то, что форматируется колонка; **D** — имя колонки; **ET** — клавиша ввода; ¤ — выбранный формат.

После выполнения команды вы увидите, что в графе D расположение чисел изменится и будет таким, как представлено в электронной таблице на стр. 133:

144.00
222.50
99.00

Для завершения первого этапа проектирования таблицы вам осталось сформировать итоговую строку.

4.2.5. ФОРМИРУЕМ ИТОГОВЫЕ РЕЗУЛЬТАТЫ

Формирование большинства таблиц заканчивается строкой, где размещаются итоги.

Проведите горизонтальную черту в электронной таблице. Для этого установите указатель в клетке A12 и введите строку

3>'=

Во всех клетках 12-го ряда появится знак =. Переведите указатель в клетку A13, переключитесь на регистр с русским алфавитом и введите строку со словом ИТОГО:

9>" **ИТОГО**

Расчет итога проведите по колонке D «ЗАРАБОТНАЯ ПЛАТА» и запишите его в клетку D13, предварительно установив там указатель. Для подсчета общей суммы заработной платы воспользуйтесь функцией суммирования СУМ и введите строку

12>СУМ (Д9:Д11)

В клетке D13 появится число 465.50. Убедитесь в правильности расчета, сложив три числа 144+222,5+99= =465,5. Выражение Д9:Д11, являющееся аргументом функции СУМ, означает, что суммируется содержимое клеток D9, D10, D11.

Пояснение к функции СУМ

Эту функцию рекомендуем применять при суммировании нескольких слагаемых. В скобках через запятую перечисляются клетки и блоки, значения которых следует просуммировать. Возможно, вам понадобится определить сумму чисел, находящихся в разных рядах и колонках. Например, для определения суммы чисел, находящихся в клетках A1, A2, A3, A4, B7, B8, C3, D4, D5, D6, достаточно ввести в соответствующую клетку функцию, используя русский алфавит:

СУМ (А1:А4, Б7, Б8, Ц3, Д4:Д6)

4.2.6. ПЕЧАТАЕМ ТАБЛИЦУ

Закончив формирование таблицы, вы хотите ее напечатать. Это возможно сделать командой **О**, где вам предоставляются две возможности:

- в соответствии с первым вариантом, названным «DAHНЫЕ», распечатывается таблица с конкретными числами;
- согласно второму варианту, названному «FOPМУЛЫ», напечатаются формулы, содержащиеся в клетках, причем

в каждой строке печати будет содержаться одна клетка с соответствующей формулой.

Наберите в строке ввода и введите

3>**/O**

Появится подсказка «ПОКАЗЫВАТЬ DАННЫЕ ИЛИ FОРМУЛЫ».

Вам нужна вся таблица. Поэтому нажмите клавишу **D**, тем самым выбрав вариант «DАННЫЕ». Появление новой подсказки «УКАЖИ МЕСТО» вынуждает вас указать конкретную область таблицы. Такой областью может служить клетка, несколько рядом стоящих клеток (блок), ряд, колонка, вся таблица. Для изображения диапазона клеток указываете первую и последнюю клетки, разделив их двоеточием. Для печати всей таблицы наберите слово ВСЕ и нажмите клавишу **ЕТ**. Таким образом, строка ввода примет вид:

22>**/ОТПЕЧАТ., DАННЫЕ, ВСЕ,**

а в строке подсказки появится новое сообщение «УКАЖИ УСТР.— ВО/РЕЖИМ:АЦПУ, NАСТРОЙКА, КОНСОЛЬ ИЛИ DИСК». Для печати на принтере достаточно нажать клавишу **А**. Через несколько секунд вы держите в руках лист с отпечатанной таблицей.

Итак, для печати на принтере всей таблицы вы нажали клавиши: **/OD ВСЕ ЕТ А**.

4.2.7. ЗАПИСЫВАЕМ ТАБЛИЦУ НА ДИСК, СЧИТЫВАЕМ ТАБЛИЦУ С ДИСКА

Запись на диск

После окончания работы с таблицей запишите ее на диск, так как выключение компьютера приводит к стиранию содержимого оперативной памяти и ваша таблица не

сохранится. Воспользуйтесь командой **S**(SOXPAH.). Введите команду **/S**. В строке подсказки появится сообщение «УКАЖИ ИМЯ ТАБЛИЦЫ (ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ)».

Задайте имя таблицы и введите его. Имя может состоять из одного — восьми символов и предшествующего имени дискового устройства с двоеточием. Присвоим нашей таблице имя ЗАРПЛАТА и запишем его на диск А. Учитывая, что ВАРИТАБ находится на том же диске, в имени таблицы можно опустить имя дисковода. Наберите слово ЗАРПЛАТА и нажмите клавишу ввода **ЕТ**. Тогда строка ввода примет вид

20>**/SOXPAHИТЬ, ЗАРПЛАТА**

В строке подсказки появится сообщение «WCE, ZHAЧЕНИЯ ИЛИ /\ АСТЬ».

Для записи всей таблицы нажмите клавишу **W**, и команда начнет выполняться. После записи таблицы на диск вы уже не зависите от компьютера. Можете спокойно вынуть диск и выключить компьютер или выйти в операционную систему командой /**QD**.

Запись в оперативную память

На следующий день у вас появилось желание вновь поработать с таблицей, зарегистрированной под именем «ЗАРПЛАТА». Включите компьютер, загрузите с диска в оперативную память ВАРИТАБ, введя имя

A>**WT**

А затем воспользуйтесь командой **Т** (ТАБЛ.) для записи таблицы с диска в оперативную память. Нажмите клавиши /**Т**, и в строке подсказки появится сообщение «УКАЖИ ИМЯ ТАБЛИЦЫ (ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ)». Наберите имя таблицы ЗАРПЛАТА и нажмите клавишу **ЕТ**. Строка ввода примет вид

17>**/ТАБЛ., ЗАРПЛАТА**

В строке подсказки появится новое сообщение «WCЮ ИЛИ /\ АСТЬ ТАБЛ.». Для переписи всей таблицы на-

жмите клавишу **W**, и команда начнет выполняться. После окончания работы команды на экране появится таблица с именем «ЗАРПЛАТА».

Возможно, что вы забыли имя таблицы. Не отчаивайтесь, это легко восстановить. После появления подсказки «УКАЖИ ИМЯ ТАБЛИЦЫ» нажмите клавишу **ЕТ**, и на экране появится меню, с помощью которого вы вызовете каталог таблиц ВАРИТАБа, нажав клавишу **W**. Ознакомившись с содержанием каталога, для выхода из него нажмите **ЕТ**, а затем одновременно клавиши **CTRL Z**, и вы вернетесь к прежнему состоянию экрана, где в строке подсказки находится

8>/ТАБЛ.,

Далее введите имя таблицы и действуйте так же, как и ранее.

4.3. ТЕМ, КТО ХОЧЕТ ЗАКРЕПИТЬ И РАСШИРИТЬ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ВАРИТАБа

План действий

Полагаем, что вы уже владеете минимально необходимыми средствами ВАРИТАБа. Тогда попробуйте полностью построить электронную таблицу, идентичную табл. 4.3. При этом вы сможете ознакомиться с рядом дополнительных средств:

- автоматической записью в диапазон клеток, идентичных по содержанию формул, но отличных по названиям переменных (ссылкам);
- созданием и использованием вспомогательных таблиц;
- редактированием предыдущей таблицы;
- использованием функций логики и поиска.

Дальнейшее изложение будем вести в более сжатой форме, расширяя его только для пояснения новых средств. Вам необходимо проделать следующее:

- дополните таблицу колонками «УДЕРЖАНИЯ»

и «СУММА К ВЫДАЧЕ», одновременно проводя их форматирование;

◆ проведите редактирование таблицы, представленной на стр. 133, вводя пустую строку и названия трех новых колонок, затем введя формулы расчета в каждую клетку этих колонок, вы получите результаты;

◆ при записи формул вам придется воспользоваться логическими функциями и вспомогательными табл. 4.1 и 4.2 расчета налогов.

Сформируйте шапку

Ширина каждой новой графы (см. табл. 4.3) равна шести позициям плюс одна позиция слева для символа :. Проведите форматирование колонок E, F, G.

Установите указатель в колонке E и введите команду форматирования, нажимая клавиши **/F К E ET 7 ET**.

Установите указатель в колонке F и введите команду

/F К F ET 7 ET

Установите указатель в колонке G и введите команду

/F К G ET 7 ET

Введите названия колонок

●●● Установите указатель в клетке E5 и введите

13>": **УДЕРЖАНИЯ**

●●● Верхнюю и нижнюю горизонтальные черты шапки вводить не надо, так как вы в предыдущем разделе ввели символ = во все клетки соответствующих рядов.

●●● Проведите горизонтальную черту в клетках E6—F6. Для ограничения горизонтальной черты справа установите указатель в клетку G6 и введите кавычки. Затем установите указатель в клетку E6 и введите

3>"=

Во всех позициях клеток E6, F6 появится символ =.

●●● Увеличьте длину шапки таблицы за счет введения дополнительной пустой строки после ряда 7. Сделайте это командой WCTABKA. Введите команду **/W R 8 ET**.

На экране после ряда 7 появится пустая строка, которая имеет номер 8, а все последующие ряды изменили прежнюю нумерацию на единицу, т. е. стали 9, 10, ...

●●● Введите названия колонок E, F. Установите указатель в клетку E7 и введите

9>″: **ПРОФ.**

Установите указатель в клетку E8 и введите

9>″:**ВЗНОСЫ**

Установите указатель в клетку F7 и введите

9>″:**ПОДОХ.**

Установите указатель в клетку F8 и введите

9>″: **НАЛОГ**

Введите название колонки G. Установите указатель в клетку G5 и введите

9>″: **СУММА**

Установите указатель в клетке G6 и введите

6>″: **К**

Установите указатель в клетке G7 и введите

9>″:**ВЫДАЧЕ**

Ввод формул в колонку «ПРОФ.ВЗНОСЫ»

Расчет профсоюзных взносов ведется по формуле $W=0{,}01Z$ (см. параграф 4.2.1), т. е. в клетки электронной таблицы должны быть записаны формулы, схожие по своему назначению, но отличные по используемым в них ссылкам. Можно ли автоматизировать процесс записи идентичных формул в электронной таблице? Да, и этому вы научитесь, если внимательно отнесетесь к тому, что изложено ниже.

Представьте себе, что у вас не три клетки, как в нашем примере, а сотни таких клеток. Набирать и вводить в каждую клетку формулу одинаковой структуры — процесс утомительный. Из этого положения есть простой выход — использовать команду размножения R. Эта команда в указанные клетки записывает требуемую формулу с одновременным изменением в ней ссылок.

Внимание! Ссылки в новых скопированных формулах отражают такое же взаимное расположение клетки-результата и клеток с исходными данными, как и в формуле-оригинале, которая копируется командой **R**.

Итак, у вас имеется клетка **E10**, в которой хранится формула-оригинал 0.01*D10. Вам надо, чтобы в клетке E11 находилась формула 0.01*D11, а в клетке E12 — 0.01* D12. Как видите, формулы различаются только ссылками. Причем указывают они на клетки, расположенные относительно результирующих клеток так же, как в формуле-оригинале. Подобное автоматическое изменение ссылок в копируемой формуле называется автоматической настройкой формулы, в отличие от индивидуальной настройки, где вам предоставляется возможность влиять на настройку, отменяя ее или разрешая.

Командой размножения **R** воспользуйтесь следующим образом:

◆ введите команду **/R**, и в строке подсказки появится сообщение «ОТКУДА? УКАЖИ МЕСТО», т. е. вам надо указать клетку, где хранится формула-оригинал,— E10;

◆ нажмите клавиши **E10 ET**, вновь появится сообщение «КУДА? УКАЖИ МЕСТО, ЗАТЕМ "ET"; или "," ДЛЯ ВАРИАНТОВ»;

◆ вам необходимо скопировать формулу с настройкой в клетки E11, E12; для этого задайте диапазон клеток, нажав клавиши **E11:E12 ET**;

◆ вы увидите, что в клетке E11 появилось число 2.225, а в клетке E12 — число 0.99;

◆ убедиться в правильности копирования вы можете, установив указатель сначала в клетку E11, а затем — в клетку E12; при этом наблюдайте за строкой состояния, где будет индицироваться содержимое этих клеток в виде формул.

Формирование вспомогательных таблиц

Вспомогательные табл. 4.1, 4.2 разместите в свободных колонках электронной таблицы: табл. 4.1 займет колонки

H, I, табл. 4.2 — колонки J, K. Для большей наглядности упростите названия таблиц и ее граф.

Установите на экране колонки H, I, J, K. Для этого переместите указатель в крайнюю правую колонку H и нажмите четыре раза клавишу **ET**.

●●● Формирование первой таблицы.

Установите указатель в клетку H1 и введите

20>"НАЛОГ ДО 100 РУБ.

Установите указатель в клетку H2 и введите

3>'=

Установите указатель в клетку H3 и введите

10>"ЗАРПЛАТА

Установите указатель в клетку H4 и введите

4>'=

Установите указатель в клетку I3 и введите

10>": НАЛОГ

Итак, вы закончили формирование заголовка и шапки первой таблицы. Перейдите к вводу данных.

Установите указатель в клетке H5 и введите 95, затем переместите указатель в клетку I5 и введите 7.60. Повторите аналогичные действия для последующих клеток H6, I6, H7, I7, ..., вводя туда числа из табл. 4.1. Результатом ваших действий будет таблица, которая изображена ниже.

```
  !   H    !!   I   !
 1!НАЛОГ ДО 100 РУБ.
 2!==================
 3!ЗАРПЛАТА :  НАЛОГ
 4!==================
 5!        95      7.6
 6!        96     7.72
 7!        97     7.84
 8!        98     7.96
 9!        99     8.08
10!       100      8.2
```

6 Заказ № 594

●●● Формирование второй таблицы.

Установите указатель в клетку J1 и введите

21>"НАЛОГ СВЫШЕ 100 РУБ.

Установите указатель в клетку J3 и введите

10>"ЗАРПЛАТА

Установите указатель в клетку К3 и введите

10>": НАЛОГ

Горизонтальную черту не вводите, так как она введена предыдущими действиями при формировании первой таблицы.

После формирования заголовка и шапки таблицы перейдите к вводу данных из табл. 4.2. Для того чтобы в дальнейшем для определения налога воспользоваться функцией ПОДБОР, следует представить числа графы «ЗАРАБОТНАЯ ПЛАТА» из табл. 4.2 в ином виде. Установите указатель в клетку J5 и введите число 101. Переведите указатель в клетку К5 и введите число 1.3. Повторите аналогичные действия для последующих пар клеток J6, К6, J7, К7, ..., вводя данные табл. 4.2. Результатом ваших действий будет таблица, изображенная ниже.

```
  !   J    !!   K    !
 1!НАЛОГ СВЫШЕ 100 Р.
 2!==================
 3!ЗАРПЛАТА :   НАЛОГ
 4!==================
 5!       101       1.3
 6!       201       2.6
 7!       301       3.9
 8!       401       5.2
 9!       501       6.5
10!       601       7.8
```

Ввод формул вычисления подоходного налога

Для вычисления подоходного налога согласно условиям, изложенным в параграфе 4.2.1, воспользуйтесь функциями. Поиск налога во вспомогательных таблицах мож-

но организовать с помощью функции ПОДБОР. Функция ИФ (если) позволит вам записать условие поиска в этих таблицах. Учитывая, что налог исчисляется исходя из целой части числа, характеризующего заработную плату, используйте функцию ЦЕЛ. С назначением и правилом использования этих функций вы можете ознакомиться в параграфе 4.4.6.

Установите указатель в клетку F10 и введите, пользуясь регистром русского алфавита,

67>ИФ (ЦЕЛ (Д10)<=100, ПОДБОР (ЦЕЛ(Д10), Х5:Х10), 8.2+ПОДБОР (ЦЕЛ(Д10), Й5:Й10))

Сейчас мы продемонстрируем еще один способ копирования клеток, удобный, когда надо изменять не все ссылки.

Установите указатель в клетку F10. Чтобы каждый раз не набирать эту формулу, в дальнейшем воспользуйтесь командой копирования (К) и редактирования (Р). Введите команду /**К F10 ЕТ F11 ЕТ**. Затем вызовите содержимое клетки F11, т. е. копию формулы, и проведите редактирование командой Р. Для этого введите команду /**Р F 11 ЕТ.** С помощью курсора исправьте все ссылки. После редактирования формула примет вид

ИФ (ЦЕЛ(Д11)<=100, ПОДБОР (ЦЕЛ(Д11), Х5:Х10), 8.2+ПОДБОР (ЦЕЛ(Д11), Й5:Й10))

Учитывая, что ВАРИТАБ во многих формулах автоматически замещает русские буквы латинскими, указанная формула идентична следующей:

IF (CEL(D11)<=100, PODBOR (CEL(D11), H5:H10), 8.2+PODBOR (CEL(D11), J5:J10))

Установите указатель в клетку F12 и проведите аналогичные действия: /**КF11 ЕТ F12 ЕТ** и /**РF12 ЕТ.** Формула, записываемая в эту клетку, примет вид

ИФ (ЦЕЛ(Д12)<=100, ПОДБОР (ЦЕЛ(Д12), Х5:Х10), 8.2+ПОДБОР (ЦЕЛ(Д12), Й5:Й10))

В процессе ввода этих формул в клетки F10, F11, F12 там появляются числа 9.5, 10.8, 8.08.

Результат, аналогичный описанному выше, вы можете получить, если воспользуетесь командой размножения R с параметром индивидуальной настройки S, где вы ответом Д(а) или Н(ет) определяете настраиваемые ссылки.

Ввод формул вычисления суммы и итога

●●● Воспользуйтесь формулой $S=Z-W$ из параграфа 4.2.1. Установите указатель в клетку G10 и введите

12>**D10—E10—F10**

Далее примените команду размножения R для записи в клетки G11, G12 идентичных по структуре формул, но содержащих отличные от исходных ссылки. Введите команду **/R G10 ET G11:C12 ET.** Результатом описанных выше действий будет появление в клетках G10, G11, G12 чисел 133.06, 209.48, 89.93.

●●● Для расчета итога по колонке G воспользуйтесь уже известной формулой вычисления суммы. Установите в клетке G14 указатель и введите

13>**СУМ (G10:G12)**

После ввода этой формулы в клетку G14 там появляется итоговый результат, равный 432.47.

Вы получили электронную таблицу *«ВЕДОМОСТЬ НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ»*

Вы проделали все основные операции по формированию электронной таблицы и заполнения ее данными. ВАРИТАБ автоматически по мере ввода данных производил перерасчет уже заполненных колонок и выводил результат в клетках. Для вас этот процесс расчета был практически незаметен. Электронная таблица по расчету заработной платы в соответствии с формой, приведенной в табл. 4.3, имеет следующий вид.

```
0 !      A         !! B !! C  !!     D     !! E  !! F  !! G  !
 1!             ВЕДОМОСТЬ
 2!      НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ
 3!
 4!=============================================================
 5!      ФАМИЛИЯ        :ТАРИФ: КОЛ. :ЗАРАБОТНАЯ:УДЕРЖАНИЯ     : СУММА
 6!  ИМЯ,ОТЧЕСТВО       :     :ОТРАБ.:  ПЛАТА   ==============:  К
 7!                     :     : ДНЕЙ :          : ПРОФ.:ПОДОХ.:ВЫДАЧЕ
 8!                                             :ВЗНОСЫ: НАЛОГ
 9!=============================================================
10!КОРНЕВ К.К.            7.2      20     144.00   1.44    9.5 133.06
11!ЛАПИН Л.Л.             8.9      25     222.50  2.225   10.8 209.48
12!МАРКОВА М.М.           5.5      18      99.00    .99   8.08  89.93
13!=============================================================
14!  ИТОГО                                465.50               432.47
```

Если вы посмотрите на эту таблицу повнимательнее, то заметите некоторые погрешности в оформлении. Исправить их достаточно легко, используя уже известные команды, например, редактирования (Р) или изменения форматов (F).

4.4. ОБЩАЯ СПРАВКА О ВАРИТАБе

4.4.1. УПРАВЛЯЮЩИЕ СИМВОЛЫ И «МЕСТО»

Вам будет интересно ознакомиться с содержанием этого параграфа только после работы на персональном компьютере РОБОТРОН 1715 по методике, изложенной в предыдущем параграфе. Сюда же вы будете обращаться и в дальнейшем для более глубокого изучения электронной таблицы. Здесь приведена справочная информация о командах и управляющих символах ВАРИТАБа. Пояснение к каждой команде, включая иллюстративный мате-

Таблица 4.4

Управляющие символы

Клавиша	Действия компьютера и пояснения
/	Начало команды. Этот символ предшествует вводу любой команды
=	Перемещение указателя в другую клетку. При нажатии этой клавиши в строке ввода появляется стрелка ==>, а в строке подсказки — сообщение «УКАЖИ, КУДА ПРЫГНУТЬ». Вам необходимо ввести номер клетки. Например, нажав клавиши ==Е3 ЕТ, вы переместите указатель в клетку Е3.
!	Режим пересчета таблицы. Этим символом пользуются после установки командой задания параметров представления и пересчета/AU режима управляемого расчета. В этом режиме пересчет таблицы осуществляется только после нажатия клавиши !. В остальных случаях клавиша ! не действует
;	Перемещение указателя между окнами. Клавишу ; следует нажимать одновременно с клавишей временного перехода на верхний регистр **ВПДР**
	Перемещение указателя <> по таблице на одну клетку
CTRL S (←)	влево
CTRL D (→)	вправо
CTRL E (↑)	вверх
CTRL X (↓)	вниз
	при условии, что не начато заполнение строки ввода. В скобках приведена идентичная клавиша для другой версии ВАРИТАБа
	Редактирование строки ввода (начато заполнение строки ввода). Четыре клавиши направления работают только в строке ввода

Клавиша	Действия компьютера и пояснения
CTRL S (←) или CTRL D (→)	Курсор перемещается влево или вправо по строке ввода. Для исправления символа достаточно установить курсор в требуемую позицию и нажать нужную клавишу
CTRL E (↑)	Вставка пробела в позиции, указанной курсором
CTRL X (↓)	Удаление позиции строки ввода, указанной курсором
CTRL Z или ⇤	Очистка строки ввода
"	Начало ввода текста в клетку
'	Начало ввода повторяющегося текста во все подряд стоящие пустые клетки начиная с текущей и в направлении, указанном в строке состояния до первой занятой клетки

риал в виде рисунков, примеров, таблиц, отражает ее основные функции.

Перечень управляющих символов ВАРИТАБа приведен в табл. 4.4.

При работе с командами вы столкнетесь с необходимостью указать «место» — клетку, ряд, колонку, несколько рядом стоящих клеток (блок), с которыми будут производиться соответствующие действия. Для указания такого места воспользуйтесь табл. 4.5.

Для управления работой электронной таблицей служат команды. Каждая команда начинается с символа / (слэж). Нажав клавишу с этим символом, вы увидите, что в строке подсказки появится сообщение

УКАЖИ А, Б, В, Г, Д, З, И, К, Л, М, Н, О, П,
Р, С, Т, У, Ф, Я ?

Вам предлагается ввести команду, нажав клавишу с первой буквой имени команды. Ввод команды вновь изменит состояние строки подсказки, где вам будет пред-

Таблица 4.5

«Место»

<table>
<tr><th>Клавиши</th><th>Наименование</th><th>Пример</th></tr>
<tr><td>Цифровые 1—254</td><td>Ряд</td><td>25—указывается 25-й ряд</td></tr>
<tr><td>Одна или две буквы от A до BK</td><td>Колонка</td><td>B — указывается колонка B (латинский алфавит)

BK — указывается колонка BK (латинский алфавит)</td></tr>
<tr><td>Буквы колонки и цифры ряда</td><td>Клетка</td><td>D15 — указывается клетка D15</td></tr>
<tr><td>Буквы и цифры : буквы и цифры</td><td>Блок —две ссылки, разделенные двоеточием</td><td><table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td>*</td><td>*</td></tr><tr><td>3</td><td></td><td>*</td><td>*</td></tr><tr><td>4</td><td></td><td>*</td><td>*</td></tr></table>B2 : C4 — указываются клетки B2, B3, B4, C2, C3, C4</td></tr>
<tr><td>ET</td><td>Текущий ряд, колонка или клетка</td><td></td></tr>
<tr><td>ВСЕ</td><td>Вся таблица</td><td></td></tr>
<tr><td>ESC</td><td>Просмотр содержимого клеток таблицы в процессе формирования строки ввода</td><td>Просмотр содержимого других клеток в процессе заполнения строки ввода. Указатель может перемещаться по таблице при частично заполненной строке ввода. Повторное нажатие ESC отменяет режим.</td></tr>
</table>

ложено уточнить содержание введенной команды за счет выбора дополнительного параметра. Введя параметр, вы можете вновь получить подсказку, и так до тех пор, пока полностью не будет сформирована команда.

Обратите внимание! Работая с электронной таблицей, пользуйтесь русским алфавитом, о чем будет свидетельствовать зажженная лампочка индикации на клавише фиксации регистров алфавита. Однако подсказка или «меню» в каждой команде будет содержать предложение о вводе параметра латинского алфавита. Причем текст этого предложения приводится на русском языке и в нем первые русские буквы каждого слова заменены на латинские буквы. Эти буквы и будут параметрами команды. Не переключайте регистр! Достаточно нажать соответствующую клавишу с латинской буквой и команда выполнится.

Например, в строке подсказки появится сообщение «BЕЗ НАСТР., S НАСТР. ИЛИ ZНАЧЕНИЯ». Нажмите одну из клавиш с латинскими буквами **B**, **S**, **Z**, не переключая регистр с русского на латинский алфавит.

4.4.2. ХАРАКТЕРИСТИКА КОМАНД

В этом и последующих параграфах вам предлагается ознакомиться с назначением и функциями каждой команды. Обратившись к табл. 4.6, вы получите общее представление о командах ВАРИТАБа для компьютера Роботрон 1715. Их всего 19 и они условно могут быть разбиты на три группы по функциональному назначению:

- команды изменения содержимого таблицы;
- команды взаимодействия таблицы с памятью и принтером;
- команды управления режимом работы и формой представления.

Для того чтобы более глубоко изучить возможности каждой команды, рекомендуем ознакомиться с содержанием последующих параграфов данной главы, где по каж-

Таблица 4.6

Команды

Команда			Назначение команды
Клавиша алфавита		Условное обозначение	
русского	латинского		
Изменение содержимого таблицы			
Б	B	ВЛАНК	Удаление содержимого клеток
В	W	WCTABKA	Ввод нового ряда или колонки
П	P	PPABKA	Редактирование содержимого клетки
М	M	МЕНЯТЬ	Перемещение колонки или ряда в другое место
К	K	КОПИЯ	Копирование содержимого клетки
Р	R	RАЗМНОЖ	Повторение части колонки или ряда
Г	G	GАРАНТ	Установка защиты клетки (запрещается ввод в эту клетку)
У	U	UБР. ГАРАНТ	Снятие защиты клетки (разрешается ввод в эту клетку)
Взаимодействие таблицы с памятью и принтером			
Н	N	NОВАЯ ТАБЛИЦА	Стирание таблицы из оперативной памяти
С	S	SОХРАН	Перезапись таблицы из оперативной памяти на диск
Т	T	ТАБЛ	Перезапись таблицы с диска в оперативную память
Л	L	LИКВИД	Стирание таблицы с диска и ряда (колонки) из оперативной памяти

Команда			Назначение команды
Клавиша алфавита		Условное обозначение	
рус-ского	латин-ского		
О	O	ОТПЕЧАТ	Печать таблицы на принтере, вывод на экран или диск
Управление режимом работы и формой представления			
Я	Q	Q КОНЧАЮ	Завершение работы
А	A	АЛЬТЕРН	Задание параметров представления и пересчета
З	Z	ZАГОЛОВКИ	Фиксация заголовка и левых колонок таблицы
Д	D	DВА ОКНА	Разделение экрана на окна
Ф	F	FОRМАТ	Изменение формата представления данных в клетках
И	I	IСПОЛНИТ	Исполнение командного файла

дой команде дается разъяснение, сопровождаемое конкретными примерами и иллюстрациями. На рисунках отображена иерархическая структура команды, состоящая из нескольких уровней. Первый уровень образуют символ / и буква команды, а последующие уровни — параметры этой команды, представленные также буквами.

Полезно вам также разобраться с примерами, которые сведены в табл. 4.7—4.10, где показано, как в соответствии с вводимыми управляющими символами, параметрами, данными изменяются состояния строк ввода и подсказки, а в некоторых случаях — и состояние экрана. Каждая команда иллюстрируется наиболее типичным вариантом

ввода. Когда реакция компьютера на ваши ответы будет неоднозначной, приводится несколько вариантов. Перед каждым примером вводите команду **/ND**.

4.4.3. КОМАНДЫ ИЗМЕНЕНИЯ СОДЕРЖИМОГО ТАБЛИЦЫ

Назначение

Группа команд B, W, P, M, K, R, G, U изменения содержимого таблицы позволяет вам отредактировать старую таблицу и сформировать новую таблицу. Вся работа по редактированию производится с текущей таблицей, находящейся в оперативной памяти.

B (BЛAHK). Удаление содержимого клеток

С помощью этой команды вы очистите «место» в таблице, т. е. клетку, ряд, колонку, блок, всю таблицу. При вводе команды подсказка напомнит вам о необходимости ввода «места», для чего воспользуйтесь рекомендациями табл. 4.5. На рис. 4.2 команда B представлена в виде двух уровней: на первом — команда, на втором — «место». Примеры использования команды B для очистки клетки Ц5 и блока А1:Ц3 приведены в табл. 4.7.

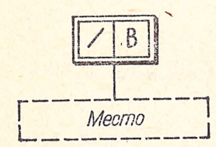


Рис. 4.2. Структура команды удаления содержимого клеток B (BЛAHK)

Рекомендуем! Для стирания всей таблицы лучше воспользоваться командой **N** (NOBAЯ ТАБЛИЦА). Команду B (BЛAHK) применяйте для очистки клеток.

W (WСТАВКА). *Ввод нового ряда или колонки*

С помощью этой команды вы можете вставить пустой ряд или колонку в любое место таблицы. При этом произойдет переименование всех рядов или колонок начиная со вставленного ряда или вставленной колонки. На рис. 4.3 команда W представлена тремя уровнями:

- на первом вводите команду;
- на втором выбираете ряд или колонку;
- на третьем конкретно указываете номер ряда или букву колонки.

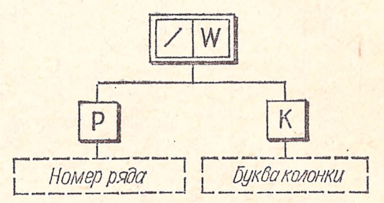


Рис. 4.3. Структура команды ввода нового ряда или колонки W (WСТАВКА)

Пример использования команды W для вставки пустой колонки B (Б) приведен в табл. 4.7. По окончании действия команды произойдет переименование колонок. Новая пустая колонка займет место между старыми колонками А и В и получит название В. Старая колонка В переименована на С и т. д.

P (PPАВКА). *Редактирование содержимого клетки*

Эта команда позволит вам внести изменения в содержимое клетки. Предварительно установите указатель в клетку, подлежащую редактированию. В противном случае, хотя по команде P в строку ввода и будет вызвано содержимое редактируемой клетки, запись отредактированного содержимого клетки будет произведена в ту клетку, где установлен указатель. Введя команду редактирования, следует ответить на предлагаемую подсказку о вводе адреса клетки или просто нажать клавишу ET для вызова содержимого текущей клетки. Затем после

Таблица 4.7

Примеры команд об изменении содержимого таблицы

Нажимаемые клавиши	Строка ввода	Строка подсказки	Пояснения
Команда B (Б)			
/B Ц5 ЕТ	/ВЛАНК /ВЛАНК, Ц5	УКАЖИ МЕСТО ЧИСТКА ПРОБЕЛОМ	После ввода команды клетка Ц5 (C5) будет пустой
/B A1 : Ц3 ЕТ	/ВЛАНК, /ВЛАНК, A1 : Ц3	УКАЖИ МЕСТО ЧИСТКА ПРОБЕЛОМ	После ввода команды блок клеток A1, A2, A3, C1, C2, C3 будет пустым
Команда W (В)			
/W	/WСТАВКА,	RЯД ИЛИ КОЛОНКУ?	Старый экран; Новый экран (см. ниже)
K	/WСТАВКА, КОЛОН.,	УКАЖИ БУКВУ КОЛОНКИ	
Б ЕТ	/WСТАВКА, КОЛОН., Б	ВСТАВКА	

Старый экран

	A	B	C	D
1	10	15	50	
2	20	25	60	
3	30	35	70	

Новый экран

	A	B	C	D
1	10		15	50
2	20		25	60
3	30		35	70

Команда Р (П)

/Р А4 ЕТ	/РРАВКА, /РРАВКА, А4	ИЗ? УКАЖИ КЛЕТКУ —	Указатель установлен в клетке А4. Вызов содержимого А4 (можно нажать только ЕТ)
	А2 + Е3 А2 + Е3/2	— —	Редактирование формулы с помощью клавиш (символов) направления
ЕТ	—	—	Запись результата редактирования

Команда М (М)

/М	/МЕНЯТЬ,	RЯД ИЛИ КОЛОНКУ?	Старый экран (см. ниже)
К	/МЕНЯТЬ, КОЛОН.,	ИЗ? УКАЖИ БУКВУ КОЛОНКИ	
А ЕТ	/МЕНЯТЬ, КОЛОН., А,	В? УКАЖИ БУКВУ КОЛОНКИ	
Ц ЕТ	/МЕНЯТЬ, КОЛОН., А, Ц	—	Новый экран (см. ниже)

Старый экран

	A	B	C
1	10	7	0.1
2	20	7	0.2
3	30	7	0.3

Новый экран

	A	B	C
1	7	0.1	10
2	7	0.2	20
3	7	0.3	30

Нажимаемые клавиши	Строка ввода	Строка подсказки	Пояснения
		Команда К (К)	
/К	/КОПИЯ,	ОТКУДА? (УКАЖИ МЕСТО)	Копирование чисел из блока в блок | A | B | C 1 | 20 | | 2 | 30 | | 20 3 | 40 | | 30 4 | | | 40
А1 : А3 ЕТ	/КОПИЯ, А1 : А3,	КУДА? (УКАЖИ КЛЕТКУ) И "ЕТ" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Ц2 ЕТ	/КОПИЯ, А1 : А3, Ц2	СН. КОПИИ	
/К	/КОПИЯ,	ОТКУДА? (УКАЖИ МЕСТО)	Копирование клетки в клетку с индивидуальной настройкой формулы | A | B | C | D 1 | | | | 5 2 | | 30 | 13 | 7 3 | 63 | 3 | | 2*B2+B3 2*D1+B3
А3 ЕТ	/КОПИЯ, А3,	КУДА? (УКАЖИ КЛЕТКУ)	
		И "ЕТ" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Ц2	/КОПИЯ,	ВЕЗ НАСТР., S НАСТР., ИЛИ ZНАЧЕНИЯ	
,	А3, Ц2,		
S	2 * <B2> + B3	ИСХОДН. КЛЕТКЕ А3 НАСТР. В2 (Д < > Н)? СН. КОПИИ	
Д	2 * B2 + <B3>	ИСХОДН. КЛЕТКЕ А3 НАСТР. В3 (Д < > Н)?	
N	—	ПЕРЕСЧЕТ ТАБЛИЦЫ	

/К	/КОПИЯ,	ОТКУДА? УКАЖИ МЕСТО	Исходное состояние
Е1 : Е2 **ЕТ**	/КОПИЯ, Е1 : Е2,	КУДА? УКАЖИ КЛЕТКУ И "ЕТ" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Е3 ЕТ	/КОПИЯ, Е1 : Е2, Е3	—	После копирования с автоматической настройкой

Исходное состояние:

C	D	E	
10	100	1000	⇐ C1*D1
20	200	4000	⇐ C2*D2
30	300		
40	400		

После копирования с автоматической настройкой:

C	D	E	
10	100	1000	⇐ C1*D1
20	200	4000	⇐ C2*D2
30	300	9000	⇐ C3*D3
40	400	16000	⇐ C4*D4

Команда R (Р)

/R	/RАЗМНОЖ.,	ОТКУДА? (УКАЖИ МЕСТО)	Размножение формулы из одной клетки в несколько клеток
А2 ЕТ	/RАЗМНОЖ., А2,	КУДА? УКАЖИ МЕСТО, ЗАТЕМ "ЕТ" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Б4 : Д4 **ЕТ**	/RАЗМНОЖ., А2 Б4 : Д4	—	

	A	B	C	D	E
1	10	20			
2	30				
3		1	2	3	4
4		3	5	7	

A1+B1 B3+C3 C3+D3 D3+E3

Нажимаемые клавиши	Строка ввода	Строка подсказки	Пояснения
/R	/RAЗМНОЖ.,	ОТКУДА? (УКАЖИ МЕСТО)	Размножение значения из одной клетки в несколько клеток
A2 ET	/RAЗМНОЖ., A2,	КУДА? УКАЖИ МЕСТО, ЗАТЕМ "ET" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Б4 : Д4 , Z	/RAЗМНОЖ., A2, Б4 : Д4	ВЕЗ НАСТР., S НАСТР. ИЛИ ZНАЧЕНИЯ —	
/R	/RAЗМНОЖ.,	ОТКУДА? (УКАЖИ МЕСТО)	Размножение части колонки в блок клеток
A1 : A3 ET	/RAЗМНОЖ., A1 : A3,	КУДА? УКАЖИ МЕСТО ЗАТЕМ "ET" ИЛИ "," ДЛЯ ВАРИАНТОВ	
Ц2 : E2 ET	/RAЗМНОЖ., A1 : A3 Ц2 : E2	—	

Команда G (Г)

/G **A2 ET**	/GAPAHT., /GAPAHT., A2	УКАЖИ МЕСТО	После установки указателя в клетке A2 в строке состояния будет ФОРМ = A3 + B1 После окончания команды Р ФОРМ = A3 + B1

Команда U (У)

/U **A2 ET**	/UБР. ГАРАНТ, /UБР. ГАРАНТ., A2	УКАЖИ МЕСТО	После окончания команды в строке состояния пропадет буква Р ФОРМ = A3 + B1

появления содержимого клетки в строке ввода можно произвести любые действия по редактированию: вставку, замену, удаление, воспользовавшись управляющими символами (клавишами) изменения направления (см. табл. 4.4). На рис. 4.4 изображена структура команды редактирования:



Рис. 4.4. Структура команды редактирования содержимого клетки Р (РРАВКА)

- на первом уровне — команда;
- на втором — ввод номера клетки.

Пример использования команды Р для редактирования содержимого клетки А4 приведен в табл. 4.7. В клетке А4 находится формула А2+Е3. Сначала установите указатель в клетке А4. После ввода команды **/Р А4 ЕТ** или **/Р ЕТ** в строке ввода появится формула, измените ее и нажмите клавишу ввода **ЕТ**, для записи формулы в ту же клетку.

М (МЕНЯТЬ). Перемещение колонки или ряда

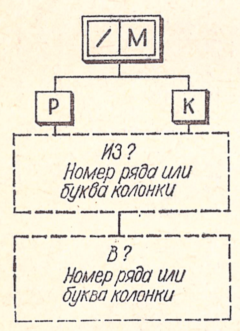


Рис. 4.5. Структура команды перемещения колонки или ряда в другое место М (МЕНЯТЬ)

С помощью этой команды вы сможете переместить в другое место таблицы содержимое колонки или ряда. Для этого укажите в команде имя перемещаемой колонки или ряда, а также имя колонки или ряда, куда производится перемещение. Невозможно поместить содержимое колонки на место ряда. При перемещении произойдет переименование колонок (рядов). На рис. 4.5 показана структура команды из четырех уровней:

- на первом уровне — команда;
- на втором уровне определяете ряд или колонку;
- на третьем уровне указываете но-

мер ряда или букву колонки, откуда надо перенести информацию;
◆ на четвертом уровне указываете номер ряда или букву колонки, куда переносится информация.

Пример перемещения содержимого колонки А в колонку С приведен в табл. 4.7, где видно, что при перемещении колонки А занятые колонки В, С сдвигаются влево, а на место колонки С встает колонка А.

К (КОПИЯ). Копирование содержимого клетки

Эта команда позволит вам снять копию содержимого клетки, блока, части ряда или части колонки и поместить ее в другое место таблицы. Причем адрес оригинала, с которого снимается копия, должен быть указан полностью, т. е. номера первой и последней клеток диапазона. Для указания адреса диапазона, куда помещается копия, достаточно ввести номер первой клетки, и команда копирования соответственно оригиналу выделит клетки для всего диапазона.

●●● Процесс копирования чисел заключается в том, что копируемые числа записываются на новом месте.

●●● Процесс копирования содержимого клеток с формулой значительно сложнее. Здесь действует механизм настройки формул. Вспомним, что в формулу наряду с константами и функциями входят переменные, которые в электронной таблице называются ссылками или адресами клеток. При копировании с нажатием клавиши ЕТ происходит автоматическое изменение ссылок. Новые ссылки будут указывать на клетки, так же симметрично расположенные относительно результирующей клетки, как и в исходной копируемой формуле-оригинале. Часто возникают ситуации, когда не все ссылки должны изменяться, некоторые должны быть неизменными. В этом случае предусмотрена индивидуальная настройка в команде копирования К с помощью параметра S.

Структуру команды копирования (рис. 4.6) образуют несколько уровней:

◆ на первом уровне вводите команду;

◆ на втором уровне указываете диапазон клеток оригинала, т. е. откуда вы хотите копировать;

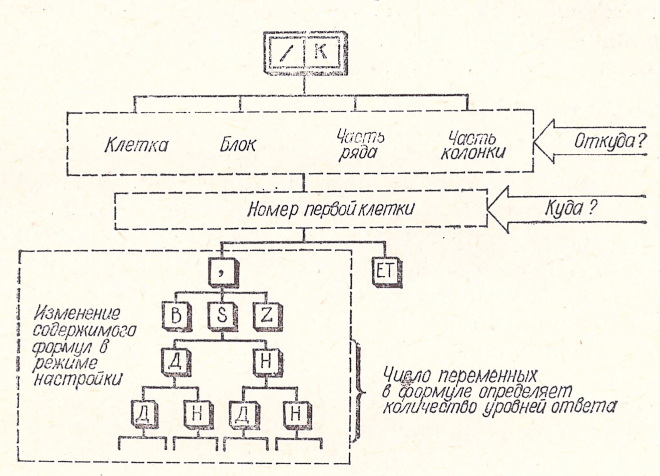


Рис. 4.6. Структура команды копирования содержимого клетки К (КОПИЯ)

◆ на третьем уровне указываете номер первой клетки диапазона, куда помещается копия;

◆ на четвертом уровне нажимаете клавишу ввода **ЕТ** при простом копировании с автоматической настройкой формул; если вы хотите произвести копирование с индивидуальной настройкой, то нажмите клавишу , и попадете на пятый уровень;

◆ на пятом уровне, называемом уровнем настройки формул, действуют три параметра: B, S, Z.

●●● Параметр B (BEZ НАСТР.) позволяет скопировать без каких-либо изменений формулы и соответствующие

им вычисленные значения. После копирования как в клетках оригинала, так и новых клетках вы увидите одно и то же число (формулу).

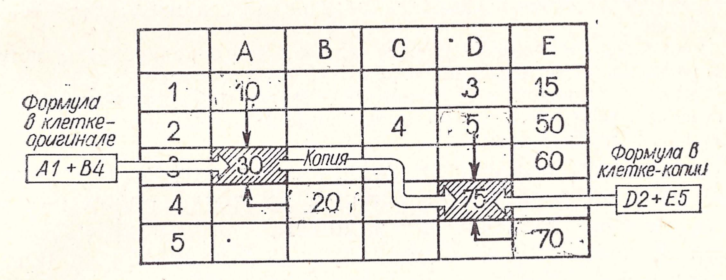
●●● Параметр Z (ZНАЧЕНИЯ) копирует только конкретные значения клеток оригинала, формулы он не копирует.

●●● Параметр S позволяет вносить изменения в копируемые формулы за счет изменения ссылок на клетку. Используя параметр S, вы сможете изменить формулу так, что не все ссылки будут автоматически изменяться, некоторые можно оставить неизменными.

Например (табл. 4.8), вы копируете клетку A3, где имеется формула A1+B4, в клетку D4, используя пара-

Таблица 4.8

Копирование содержимого клетки A3 в клетку D4 с настройкой всех ссылок в копируемой формуле



метр настройки S для ссылок A1 и B4. Тогда в клетку D4 запишется формула D2+E5, поскольку эти ссылки расположены так же симметрично относительно клетки-копии D4, как и ссылки в клетке-оригинале A3. Настройка формул производится последовательно по каждой ссылке. В строку ввода будет вызвана формула, и указатель устанавливается сначала на первой ссылке, затем на второй.

Строка подсказки приглашает вас указать ссылку, которую вы настраиваете. Путем введения ответа Д (ДА)

или Н (НЕТ) осуществляется преобразование копируемой формулы. Вы вводите Д или Н столько раз, сколько в формуле имеется ссылок.

В табл. 4.7 приведены примеры копирования чисел блока A1:A3, копирования формулы клетки A3 с индивидуальной настройкой и копирования формул из блока E1:E2 в блок E3:E4 с автоматической настройкой.

R (RАЗМНОЖ.). *Повторение части колонки или ряда*

Советуем вам ознакомиться с этой командой после изучения команды копирования содержимого клетки (К). Эти команды очень схожи между собой по структуре образования, вариантам подсказок и вводимым параметрам ответа. Однако надо основательно разобраться, в каких ситуациях использовать ту или другую команду.

Обратите внимание! Основная разница этих команд заключается в следующем:

- команда копирования К служит для обычного копирования, которое можно представить соотношением 1:1;
- команда размножения R копирует «место» таблицы в несколько аналогичных мест, что можно представить соотношением 1:N (один оригинал и несколько копий).

Так же как и команда копирования, команда размножения может делать одну копию, но тогда рекомендуем все же воспользоваться командой К. В команде размножения R существуют ограничения на диапазон копирования:

- можно размножать клетку, часть колонки или ряда в клетку, часть колонки или ряда, ряд, колонку;
- нельзя размножать блок, в этом случае появляется сообщение «ОШИБКА МЕСТА»;
- нельзя размножать колонку в колонку и ряд в ряд, на экране появится сообщение «В ДОЛЖНО БЫТЬ ЧАСТЬ РЯДА (КОЛОНКИ)».

На рис. 4.7 показана структура команды размножения, имеющая столько же уровней, сколько и в команде копи-

рования. При внимательном рассмотрении структуры вы обратите внимание на различие в формировании диапазона копирования «ОТКУДА» и «КУДА». Все нижеследующие уровни идентичны как по форме, так и по содержанию. Для их понимания обратитесь вновь к команде копирования.

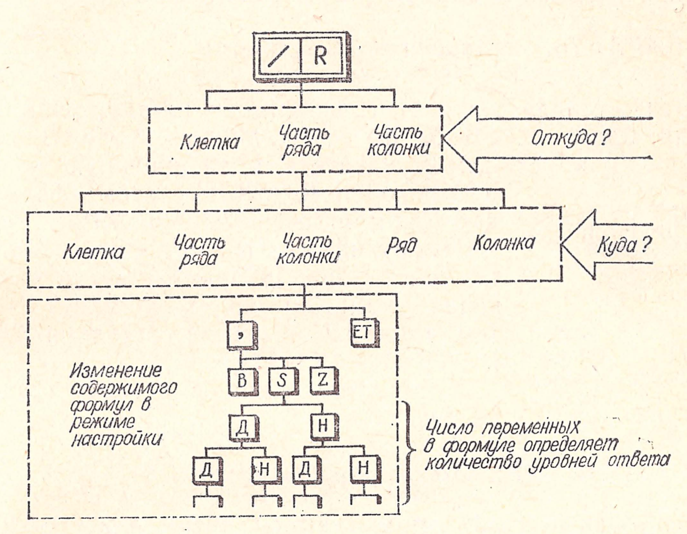


Рис. 4.7. Структура команды повторения части колонки или ряда R (RАЗМНОЖ.)

В табл. 4.7 приведены три примера, в которых показано, как размножить по команде размножения:

- формулу из одной клетки с автоматической настройкой ссылок;
- число из одной клетки в блок клеток;
- числа из части колонки в блок клеток.

Замечание! При размножении формул действуют те же правила настройки, что и для команды **К (КОПИЯ)**. Задав область, куда будет помещена копируемая формула, вы можете воспользоваться любым из вариантов копирования, нажав клавиши:

ET — произойдет автоматическая настройка формул;

B — формулы и значения скопируются без изменения;

Z — скопируются только значения;

S — часть параметров вы можете скопировать как при автоматической настройке, а часть оставить без изменения.

G (GAPAHT). *Установка защиты клетки*

Эта команда защитит содержимое клетки, ряда, колонки, блока, всей таблицы от случайного ввода. В дальнейшем при установке указателя в защищенную клетку в строке состояния появятся формула или текст, помеченные буквой Р.

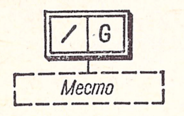


Рис. 4.8. Структура команды установки защиты клетки G (GAPAHT.)

Эта буква — сигнал о наличии защиты данной клетки. Например, после применения команды G для клетки A2 (см. табл. 4.7) в строке состояния будет Р ФОРМ=A3+B1.

Структура команды G представлена на рис. 4.8.

U (UБР.ГАРАНТ). *Снятие защиты клетки*

С помощью этой команды вы снимите защиту с клетки, ряда, колонки, блока, всей таблицы. После выполнения команды при установке указателя в ранее защищенную клетку вы увидите в строке состояния формулу или текст без предшествующей буквы Р. Структура команды U представлена на рис. 4.9. Пример использования команды U для снятия защиты с клетки A2 приведен в табл. 4.7.

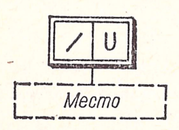


Рис. 4.9. Структура команды снятия защиты клетки U (UБР. ГАРАНТ.)

4.4.4. КОМАНДЫ ВЗАИМОДЕЙСТВИЯ ТАБЛИЦЫ С ПАМЯТЬЮ И ПРИНТЕРОМ

Назначение

Эти команды позволяют вам иметь доступ к дискам и принтеру, а также работать с таблицей как с единым целым, не занимаясь ее формированием. Основное назначение этих команд — организация связи между оперативной памятью и внешними устройствами (дисководом и принтером). Таблицу или часть таблицы можно записать на диск или отпечатать в нужном формате. Можно вызвать таблицу с диска в оперативную память или стереть ее из оперативной памяти.

N (*NОВАЯ ТАБЛИЦА*). *Стирание таблицы из оперативной памяти*

Эта команда позволит вам очистить оперативную память и уничтожить находящуюся в ней текущую таблицу. На вопрос подсказки «СТЕРЕТЬ ТАБЛИЦУ ИЗ ПАМЯТИ?» вы ответите нажатием клавиши с буквами **Д** (ДА) или **Н** (НЕТ). Будьте внимательны, не сохраненную на диске таблицу вы потеряете безвозвратно. Структура команды очень проста (рис. 4.10). Пример использования этой команды приведен в табл. 4.9.

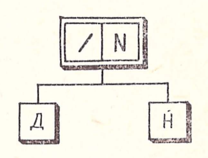


Рис. 4.10. Структура команды стирания таблицы из оперативной памяти N (NОВАЯ ТАБЛИЦА)

Таблица 4.9

Примеры команд взаимодействия таблицы с памятью и принтером

Нажимаемые клавиши	Строка		Пояснения
	ввода	подсказки	
		Команда N (Н)	
/N	/Nовая таблица?	СТЕРЕТЬ ТАБЛИЦУ ИЗ ПАМЯТИ? DA/NET	—
Д	—	—	Очистка экрана и оперативной памяти
		Команда S (С)	
/S	/SОХРАНИТЬ,	УКАЖИ ИМЯ ТАБЛИЦЫ ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ	Запись всей таблицы на диск
КОТ ЕТ	/SОХРАНИТЬ, КОТ	WСЕ, ZНАЧЕНИЯ ИЛИ ЛАСТЬ?	
W	—	НА ДИСК...	
/S	/SОХРАНИТЬ,	УКАЖИ ИМЯ ТАБЛИЦЫ ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ	Запись значения клетки А3 из текущей таблицы (на экране) в клетку А3 таблицы с именем «КОТ», находящейся на диске.
КОТ ЕТ	/SОХРАНИТЬ, КОТ,	ТАКАЯ ТАБЛИЦА УЖЕ ЕСТЬ С(МЕНИТЬ) ИМЯ, Д(УБЛЬ), П(ЕРЕЗАПИСЬ)	

Д	/SOXPAНИТЬ, КОТ	WCE, ZНАЧЕНИЯ ИЛИ ΛАСТЬ?	Причем при записи создается дубль таблицы, который содержит только значение этой клетки. Параметр WCE верхнего уровня записывает содержимое всей таблицы на диск. Параметр WCE нижнего уровня служит для записи содержимого из заданной части таблицы
Λ	/SOXPAНИТЬ, КОТ, ЧАСТ,	WCE ИЛИ ZНАЧЕНИЯ?	
Z	/SOXPAНИТЬ, КОТ, ЧАСТ, ZНАЧЕН.,	ОТКУДА? (УКАЖИ МЕСТО)	
АЗ ЕТ	/SOXPAНИТЬ, КОТ, ЧАСТ, ZНАЧЕН., АЗ	В ПАМЯТЬ	

Команда Т (Т)

/Т	/ТАБЛ.,	УКАЖИ ИМЯ ТАБЛИЦЫ (ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ)	Вызов таблицы с именем «РОЗА» с диска в оперативную память и на экран
РОЗА ЕТ	/ТАБЛ., РОЗА,	WСЮ, ΛАСТЬ, ТАБЛ?	
W	/ТАБЛ., РОЗА, WСЮ	В ПАМЯТЬ...	
/Т	/ТАБЛ.,	УКАЖИ ИМЯ ТАБЛИЦЫ (ИЛИ "ЕТ" ДЛЯ ОГЛАВЛЕНИЯ)	Перед вызовом таблицы вы знакомитесь с каталогом ВАРИТАБа, находите имя таблицы и вызываете ее в оперативную память и на экран
ЕТ	/ТАБЛ.,	УКАЖИ: S — ВЫБОР УСТР., D — ДИСК, W — ВАРИТАБ	
W	/ТАБЛ.	На экране появляется каталог таблиц ВАРИТАБа	

Нажимаемые клавиши	Строка		Пояснения
	ввода	подсказки	
ET	/ТАБЛ.,		
CTRL Z	/ТАБЛ.,		
КОТ ET	/ТАБЛ., КОТ,	WСЮ, ΛАСТЬ ТАБЛ.	
W	/ТАБЛ., КОТ, WСЮ	В ПАМЯТЬ...	
		Команда L (Л)	
/L	/LИКВИД.,	RЯД, КОЛОНКУ ИЛИ ТАБЛИЦУ?	Стирание всей таблицы с диска
T	/LИКВИД., ТАБЛ.,	УКАЖИ ИМЯ ТАБЛИЦЫ (ИЛИ "ET" ДЛЯ ОГЛАВЛЕНИЯ)	
КОТ ET	—	УДАЛЕНИЕ...	
/L	/LИКВИД.,	RЯД, КОЛОНКУ ИЛИ ТАБЛИЦУ?	Стирание колонки A из текущей таблицы оперативной памяти. На диске все осталось без изменения
K	/LИКВИД., КОЛОНК.,	УКАЖИ БУКВУ КОЛОНКИ	
A ET			

Команда O (O)

/O	/ОТПЕЧАТ.,	ПОКАЗЫВАТЬ DАННЫЕ ИЛИ FОРМУЛЫ?	Печать таблицы на принтере
D	/ОТПЕЧАТ., DАННЫЕ	УКАЖИ МЕСТО	
ВСЕ ЕТ	/ОТПЕЧАТ., DАННЫЕ, ВСЕ,	УКАЖИ УСТРОЙСТВО/ РЕЖИМ: АЦПУ, NАСТРОЙКА, КОНСОЛЬ ИЛИ DИСК	Для остановки печати таблицы нажмите **CTRL Z**
A	—	КОНЕЦ ПЕЧАТИ ТАБЛ. НАЖМИТЕ ЛЮБУЮ КЛАВИШУ	

S (SOXPAH.). *Перезапись таблицы из оперативной памяти на диск*

Этой командой вы пользуйтесь всегда, когда заканчиваете работу с таблицей. Она организует запись на диск таблицы или ее части, а также при желании — только вычисленных значений таблицы. По окончании процедуры записи таблица сохраняется в оперативной памяти и выводится на экран.

Возможна ситуация, что после запроса «УКАЖИ ИМЯ ТАБЛИЦЫ» вы случайно ввели уже имеющееся в каталоге имя. Тогда в строке подсказки появится сообщение «ТАКАЯ ТАБЛИЦА УЖЕ ЕСТЬ. С (МЕНИТЬ) ИМЯ Д (УБЛЬ), П (ЕРЕЗАПИСЬ)» и вы вводите ответ:

- введя параметр С, вы вновь окажетесь в положении, когда в строке подсказки будет предложено ввести имя таблицы; вы должны исправить или заменить уже набранное имя;
- после ввода параметра Д на диск запишется таблица, имя которой будет таким же, как у таблицы, записанной ранее; однако если вы посмотрите каталог, то увидите, что предыдущая таблица имеет расширение .DBL, а новая таблица — расширение .TBL; в дальнейшем при вызове с диска таблицы с этим именем в оперативную память и на экран попадает таблица, записанная последней и имеющая расширение .TBL; для вызова предыдущей таблицы необходимо указать ее полное имя с расширением, например, PRIMER .DBL;
- после ввода параметра П вся таблица, которая будет в дальнейшем создана на экране, запишется под тем же именем, что и предыдущая; старая таблица, хранящаяся на диске, будет стерта.

Предположим, что, прежде чем записать таблицу на диск, вы хотели бы ознакомиться с каталогом файлов. Для этого достаточно нажать после ввода команды /S на клавишу ввода **ET** и на экране появится меню:

ПЕРЕЧЕНЬ ТАБЛИЦ ВАРИТАБ С ДИСКА
ТЕКУЩЕЕ ДИСК. УСТР-ВО А
НАЗВАНИЕ ТЕКУЩЕЙ ТАБЛИЦЫ А: (ИМЯ ТАБЛИЦЫ)
ВАРИАНТ:
S — ВЫБОР ДРУГОГО ДИСК. УСТР-ВА
D — ОГЛАВЛЕНИЕ ВСЕХ ФАЙЛОВ ДИСКА
W — ТОЛЬКО ТАБЛИЦЫ, СОЗДАННЫЕ ВАРИТАБ
CTRL Z — ДЛЯ ВОЗВРАТА К ТАБЛИЦЕ

Вслед за меню в строке подсказки будет сообщение:

«УКАЖИ: S — ВЫБОР УСТР., D — DИСК,
W — ВАРИТАБ»

Вы воспользуетесь одним из перечисленных параметров:

S разрешает перейти на другой дисковод;

D вызывает каталог диска.

Обратите внимание! Все файлы, хранящие таблицы, относятся к типу .TBL. Доступ к ним с раскрытием их содержимого возможен только командами ВАРИТАБа;

W вызывает на экран только названия таблиц, созданных в среде ВАРИТАБа;

CTRL Z (одновременное нажатие этих клавиш) возвращает на экран текущую таблицу.

Итак, после ввода имени таблицы вы должны решить вопрос относительно того, что следует переписать на диск. Рассмотрите внимательно рис. 4.11, где показана структура команды S.

●●● На первом уровне вы вводите команду.

●●● На втором уровне вы вводите имя таблицы, при необходимости воспользовавшись меню (клавиша **ET**).

●●● На третьем уровне вы сообщите с помощью параметров, что следует записать на диск:

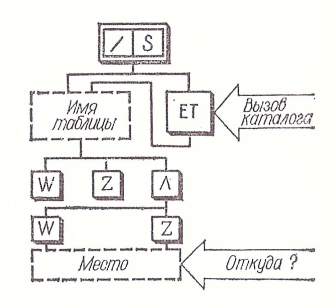


Рис. 4.11. Структура команды перезаписи таблицы из оперативной памяти на диск S (SOXPAH.)

W (WCE) — записывается вся таблица;

Z (ZНАЧЕНИЕ) — записываются только конкретные значения, находящиеся в клетках;

Λ (ΛАСТЬ ТАБЛ.) записывается часть таблицы.

Обратите внимание! Ввод этого параметра возможен только после перехода на нижний регистр. Одновременно нажмите клавишу Λ и клавишу **ВПДР**, в противном случае этот параметр не вводится.

●●● Последующие уровни уточняют смысл введенного параметра для записи части таблицы.

Для лучшего понимания работы команды S ознакомьтесь с примерами, приведенными в табл. 4.9. В первом примере показано, как на диск записать всю таблицу с именем «КОТ». Второй пример отражает более сложную структуру команды:

◆ запись таблицы производится под тем же именем, что и у предыдущей таблицы, — «КОТ»; при этом предыдущая таблица получает расширение .DBL (дубль), а новая — расширение .TBL;

◆ переписывается только значение клетки А3.

T (ТАБЛ.). *Перезапись таблицы с диска в оперативную память*

Вы уже освоили основные средства ВАРИТАБа и хотите внести некоторые изменения в ранее созданную и записанную на диске таблицу. Вызвать эту таблицу с диска в оперативную память и на экран может команда Т, в которой обязательно надо указать ее имя. Если вы забыли имя таблицы, воспользуйтесь после ввода команды /Т клавишей **ЕТ**. После ее нажатия на экране появится меню, с помощью которого можно вызвать каталог диска и найти имя.

Не забудьте! Перед вызовом таблицы следует очистить оперативную память (экран). Иначе на экран попадет «мусор», т. е. часть предыдущей таблицы, которая не заменилась.

Структура команды Т (рис. 4.12) и, соответственно, ваши действия будут аналогичными команде S. Подсказка поможет вам вызвать не только всю таблицу (W), но и ее часть (Λ). При вызове части таблицы с помощью подсказок вы уточняете, в каком виде, откуда и куда следует передать данные.

В табл. 4.9 приведен пример перезаписи всей таблицы с именем «РОЗА» с диска в оперативную память. Заметим, что на диске зафиксировано имя файла этой таблицы — «ROZA». Пользуясь регистром русского алфавита, вы нажимаете соответствующие клавиши, поэтому имя будет введено русскими буквами — РОЗА.

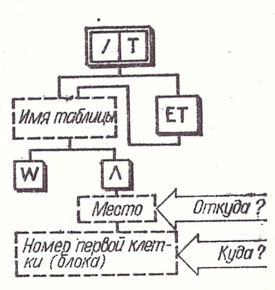


Рис. 4.12. Структура команды перезаписи таблицы с диска в оперативную память Т (ТАБЛ.)

Второй пример (см. табл. 4.9) показывает, как перезаписать всю таблицу, если вы забыли ее имя и пользуетесь каталогом ВАРИТАБа. После просмотра каталога для возврата к таблице одновременно нажмите клавиши CTRL Z.

L (LИКВИД.). *Стирание таблицы с диска и ряда (колонки) из оперативной памяти*

Эта команда обеспечит вам стирание таблицы с диска. Кроме того, она удаляет ряд или колонку из оперативной памяти и производит перенумерацию оставшихся колонок и рядов. Структура команды L представлена на рис. 4.13:

- на первом уровне вводите команду;
- на втором уровне определяете, что необходимо удалить;
- на третьем уровне указываете конкретный адрес.

В табл. 4.9 показано, как удалить таблицу с диска и колонку из текущей таблицы.

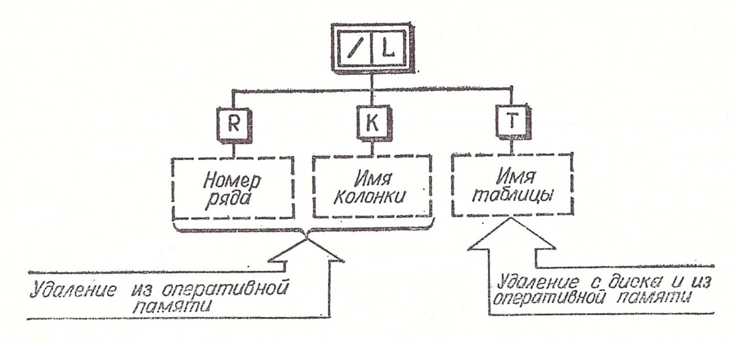


Рис. 4.13. Структура команды стирания таблицы с диска L (LИКВИД.)

О (ОТПЕЧАТ.). *Печать таблицы на принтере, вывод на экран или диск*

Эта команда позволит вам отпечатать таблицу или ее часть (блок, клетку, ряд, колонку). При необходимости можно вывести таблицу или ее часть на экран дисплея или диск. Структура команды О представлена на рис. 4.14:

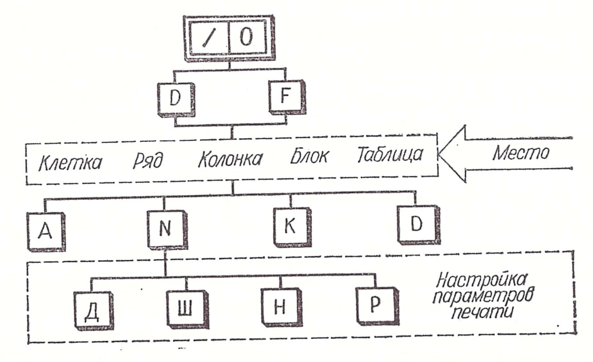


Рис. 4.14. Структура команды печати таблицы О (ОТПЕЧАТ.)

- на первом уровне вводите команду;
- перейдя на второй уровень, вы увидите подсказку «ПОКАЗЫВАТЬ DАННЫЕ ИЛИ FОРМУЛЫ?».

Если вы выберете «DАННЫЕ», то напечатается таблица, клетки которой будут заполнены конкретными числами или текстом. Если вы выберете «FОРМУЛЫ», то на каждой строке напечатаются номер клетки и находящаяся там формула. Не удивляйтесь при появлении в строке печати чисел, так как константы тоже воспринимаются как формулы;

- на третьем уровне указываете область выводимой таблицы, т. е. клетку, колонку, ряд, блок или всю таблицу;
- на четвертом уровне с помощью подсказки «УКАЖИ УСТРОЙСТВО /РЕЖИМ: АЦПУ, NАСТРОЙКА, КОНСОЛЬ ИЛИ DИСК» выберите режим.

При вводе параметров А, К, D осуществляется вывод таблицы. При вводе параметра N на экране появляется меню настройки печати:

НАСТР. ПЕЧАТИ:
Д — ИЗМЕНЕНИЕ ДЛИНЫ ЛИСТА (ДЛИНА=0 ДЛЯ НЕПРЕРЫВНОЙ ПЕЧАТИ) (ТЕК. 66 СТРОК)
Ш — ИЗМЕНЕНИЕ ШИРИНЫ ЛИСТА (ТЕК. 132 СИМВОЛА)
Н — КОДЫ РУЧНОЙ НАСТРОЙКИ
Р — ПЕЧАТЬ ТАБЛИЦЫ
CTRL Z — СНЯТЬ КОМАНДУ /О

Пример печати всей таблицы приведен в табл. 4.9.

4.4.5. КОМАНДЫ УПРАВЛЕНИЯ РЕЖИМОМ РАБОТЫ И ФОРМОЙ ПРЕДСТАВЛЕНИЯ ТАБЛИЦЫ

Q (*КОНЧАЮ*). *Завершение работы*

Эта команда позволит вам закончить работу с электронной таблицей и выйти в операционную систему. После завершения команды экран очищается и на нем появится подсказка операционной системы A>

Пример использования команды завершения работы Q приведен в табл. 4.10.

А (АЛЬТЕРН.). Задание параметров представления и пересчета

В этой команде задание одного из параметров позволит вам организовать удобный режим работы с таблицей. Параметры можно задавать, используя как русский, так и латинский алфавит. Структура команды представлена на рис. 4.15 в виде двух уровней:

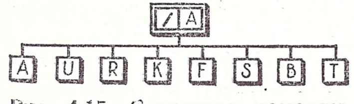


Рис. 4.15. Структура команды задания параметров представления и пересчета таблицы А (АЛЬТЕРН.)

- на первом уровне вы вводите команду;
- на втором уровне задаете режим работы с таблицей.

Всего предусмотрены восемь режимов: A, U, R, K, F, S, B, T (латинский алфавит) или А, У, Р, К, Ф, С, Б, Т (русский алфавит). Выбрав режим, нажмите соответствующую буквенную клавишу. Для отмены этого режима вы должны вновь ввести команду А и нажать клавишу другого режима.

Ознакомьтесь с первыми четырьмя режимами A, U, R, K для определения порядка проведения расчетов в таблице.

●●● Команда /АА. Режим А (АВТО) установлен по умолчанию и позволяет производить автоматический пересчет значений таблицы по мере ввода каждого данного или команды. Однако в больших таблицах из-за этого возникает большая задержка во времени, что становится помехой в работе.

●●● Команда /AU. В режиме U (U КАЗ) при вводе данных и команд пересчета таблицы не производится. Рекомендуем этот режим при вводе больших массивов данных. После окончания ввода данных достаточно нажать клавишу ! и начнется пересчет таблицы.

●●● Команда /AR. Режим R (RЯД) позволит вам организовать пересчет таблицы последовательно по рядам начиная с первого ряда, затем — второго и т. д.

●●● Команда /AK. Режим K (KОЛ.) позволит вам организовать пересчет таблицы последовательно по колонкам начиная с колонки A, затем B и т. д.

Четыре параметра F, S, B, T организуют режимы работы внутри таблицы.

●●● Команда /AF. Режим F (FОРМУЛ) обеспечит вывод в клетках таблицы текста и формул.

●●● Команда /AS. Режим S (SЛЕД) обеспечит перемещение указателя после нажатия клавиши **ET** в следующую клетку выбранного направления.

●●● Команда /AB. Режим B (BОРТ) устанавливает и снимает отображение рамок таблицы.

●●● Команда /AT. Режим T (TАБУЛ) передвигает указатель только по незащищенным клеткам, пустые и защищенные клетки перескакивают.

В табл. 4.10 показаны примеры команд снятия рамок с таблицы (режим B) и отражения в клетках таблицы формул и текста (режим F).

Z (ZАГОЛОВКИ). Фиксация заголовков и левых колонок таблицы

Этой командой удобно пользоваться при просмотре содержимого больших таблиц, занимающих несколько экранов. Для этого верхние ряды или левые колонки оставляют на экране, а все остальные ряды или колонки начинают перемещать. Например, фиксируя шапку таблицы, вы сможете просмотреть содержимое всех колонок до конца и при этом на экране будут всегда находиться названия колонок.

Прежде чем использовать эту команду, надо установить указатель. Для фиксации нескольких верхних рядов указатель установите в последнем фиксируемом ряду, а для фиксации нескольких левых колонок — в крайней правой фиксируемой колонке. Затем можно ввести команду.

Таблица 4.10

Примеры команд управления режимом работы и формой представления таблицы

Нажимаемые клавиши	Строка		Пояснения
	ввода	подсказки	
		Команда (Я)	
/Q	/Q КОНЧАЮ	ВЫЙТИ В ОП. СИСТ.? ДА или NET	—
Д	—	—	На экране появляется А >
		Команда А (А)	
/A	/АЛЬТЕРН.,	FOPМУЛ., SЛЕД., ВОРТ, ТАБУЛ., СЧЕТ: RЯД/КОЛ, UКАЗ./АВТО.?	Снятие с таблицы рамок с номерами колонок и рядов. Для восстановления рамок вновь вводится эта команда
В	/АЛЬТЕРН., ВОРТ	—	
/A	/АЛЬТЕРН.	FOPМУЛ.. SЛЕД., ВОРТ, ТАБУЛ., СЧЕТ: RЯД/КОЛ, UКАЗ./АВТО.?	В клетках таблицы показаны формулы и текст
F	/АЛЬТЕРН., FОРМУЛ	—	

Команда Z

<table>
<tr><td>/Z</td><td>/ZАГОЛОВКИ</td><td>WЕРХНЯЯ, LЕВАЯ, ОБЕ или SНЯТЬ ШАПКИ?</td><td rowspan="2">
До фиксации

<table>
<tr><td></td><td>A</td><td>B</td></tr>
<tr><td>1</td><td>ГОД</td><td>ВЫПУСК</td></tr>
<tr><td>2</td><td></td><td>МАШИН</td></tr>
<tr><td>3</td><td>=====</td><td>=====</td></tr>
<tr><td>4</td><td>1959</td><td>500</td></tr>
<tr><td>5</td><td>1960</td><td>600</td></tr>
<tr><td>6</td><td>1962</td><td>450</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>20</td><td>1985</td><td>900</td></tr>
</table>

После фиксации и просмотра рядов

<table>
<tr><td></td><td>A</td><td>B</td></tr>
<tr><td>1</td><td>ГОД</td><td>ВЫПУСК</td></tr>
<tr><td>2</td><td></td><td>МАШИН</td></tr>
<tr><td>3</td><td>=====</td><td>=====</td></tr>
<tr><td>7</td><td>1965</td><td>300</td></tr>
<tr><td>8</td><td>1966</td><td>400</td></tr>
<tr><td>9</td><td>1970</td><td>700</td></tr>
<tr><td>.</td><td></td><td></td></tr>
<tr><td>.</td><td></td><td></td></tr>
<tr><td>.</td><td></td><td></td></tr>
<tr><td>23</td><td>1988</td><td>800</td></tr>
</table>
</td></tr>
<tr><td>W ET</td><td>—</td><td>—</td></tr>
</table>

Нажимаемые клавиши	Строка ввода	Строка подсказки	Пояснения
		Команда D (Д)	
/D	/Dва окна	ДЕЛЕНИЕ: GОРИЗ., WЕРТ., ОДНО; ДВИЖЕНИЕ: SИНХРОН., АСИНХР.	Вначале установите указатель в колонке D
W	—	—	Экран после разделения по вертикали

	A	B	C		D	E	F
1	10			1	⟨50⟩		
2	20		40	2		60	
3		30		3	70		

Формулы: 2*A1, A1+A2, A1+B3

Нажимаемые клавиши	Строка ввода	Строка подсказки	Пояснения
;	—	—	Указатель переместился из колонки D в колонку C
/A	/АЛЬТЕРН.,	FОРМУЛ., SЛЕД., ВОРТ, ТАБУЛ, СЧЕТ: RЯД/КОЛ, UКАЗ/АВТО.?	Вывод формул в левом окне
F	—	—	
;	—	—	Перемещение указателя из колонки C в колонку D

=	==>	УКАЖИ, КУДА ПРЫГНУТЬ...	Установка в правом окне колонок A, B, C (см. таблицы ниже)
A ЕТ	=> A	—	
/D	/DВА ОКНА	ДЕЛЕНИЕ: GОРИЗ, WЕРТ., ОДНО; ДВИЖЕНИЕ: SИНХРОН., АСИНХР.	
S	—	—	

	A	B	C
1	10		
2	2*A1		A1+B3
3		A1+A2	

	A	B	C	D
1	<10>			50
2	20		40	
3		30		70

Команда F (Ф)

/F	/FОРМАТ,	УКАЖИ ОБЛАСТЬ: ТАБЛИЦА, КОЛОНКА, RЯД, ЭЛЕМЕНТ	
К	/FОРМАТ, КОЛОН.	УКАЖИ БУКВУ КОЛОНКИ	
Д ЕТ	/FОРМАТ, КОЛОН., Д	ЗАДАЙ ФОРМАТЫ (C, A, N, ¤, P, L, TP, TL, *, I, ШИР. КОЛОНКИ)	
¤ ЕТ	/FОРМАТ, КОЛОН., Д, ¤		

Исходное состояние

...	D	...
	2.785	
	300	
	5.8	
	736.25	

После форматирования

...	D	...
	2.79	
	300.00	
	5.80	
	736.25	

<table>
<tr><th rowspan="2">Нажимаемые клавиши</th><th colspan="2">Строка</th><th rowspan="2">Пояснения</th></tr>
<tr><th>ввода</th><th>подсказки</th></tr>
<tr><td>/F</td><td>/FОРМАТ,</td><td>УКАЖИ ОБЛАСТЬ: ТАБЛИЦА, КОЛОНКА, RЯД, ЭЛЕМЕНТ</td><td rowspan="2">Исходное состояние
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>4</td><td>5</td></tr><tr><td>2</td><td>7</td><td>1</td></tr><tr><td>3</td><td>2</td><td>3</td></tr></table></td></tr>
<tr><td>Э</td><td>/FОРМАТ, ЭЛЕМ.,</td><td>УКАЖИ МЕСТО</td></tr>
<tr><td>A1:Б3 ET</td><td>/FОРМАТ, ЭЛЕМ., A1:Б3</td><td>ЗАДАЙ ФОРМАТЫ (C, A, N, ¤, P, L, TP, TL, *, I)</td><td rowspan="2">После форматирования
<table><tr><th></th><th>A</th><th>B</th></tr><tr><td>1</td><td>****</td><td>*****</td></tr><tr><td>2</td><td>*******</td><td>*</td></tr><tr><td>3</td><td>**</td><td>***</td></tr></table></td></tr>
<tr><td>*ET</td><td>/FОРМАТ, ЭЛЕМ., A1:Б3, *</td><td></td></tr>
</table>

Команда предусматривает четыре варианта работы, что задается параметрами W, L, O, S (рис. 4.16).

●●● Команда /ZW. Параметр W (WЕРХНЯЯ) фиксирует текущий ряд и выше.

●●● Команда /ZL. Параметр L (LЕВАЯ) фиксирует текущую колонку и все левее от нее.

●●● Команда /ZO. Параметр O (ОБЕ) фиксирует одновременно верхние ряды и левые колонки.

●●● Команда /ZS. Параметр S (SНЯТЬ ШАПКИ) отменяет любую фиксацию.

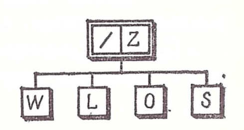


Рис. 4.16. Структура команды фиксации заголовков и левых колонок таблицы Z (ZАГОЛОВКИ)

Обратите внимание! Если предыдущая команда фиксировала верхние ряды, а вы вводите команду фиксации левых колонок, то автоматически снимается фиксация верхних рядов, и наоборот.

В табл. 4.10 приведена команда фиксации заголовка таблицы, состоящей из трех рядов. Введите заголовок таблицы и установите указатель в ряду 3. Введите команду /ZW. Переведите указатель вниз до 20 ряда и нажмите еще три раза клавишу **ET**. На экране вы наблюдаете передвижение рядов, а затем установится последний ряд с номером 23.

D (DВА ОКНА). *Разделение экрана на окна*

С помощью этой команды вы сможете разделить экран на две части (окна) либо по горизонтали, либо по вертикали. Используйте эту команду тогда, когда вам необходимо организовать одновременный просмотр разных частей электронной таблицы.

Граница между окнами определяется колонкой, где указана нумерация рядов, и положением указателя в таблице. При горизонтальном делении все ряды, расположенные вверх от указателя, образуют верхнее окно. Все ряды вниз от указателя, а также текущий ряд, где находится указатель, образуют нижнее окно.

При вертикальном делении все колонки, размещенные справа от указателя, а также текущая колонка, где находится указатель, образуют правое окно, а все колонки, расположенные слева от указателя, — левое окно.

После разделения экрана на два окна перемещать указатель между окнами можно с помощью клавиши ;. Не забудьте нажать ее одновременно с клавишей **ВПДР.** Внутри окна перемещение указателя производится обычным способом с помощью клавиш (символов) направления.

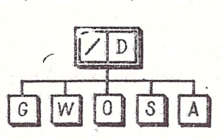


Рис. 4.17. Структура команды разделения экрана на окна D (DВА ОКНА)

В команде разделения экрана (рис. 4.17) используются параметры G, W, O для выбора варианта деления на окна, а параметры S, A — для организации режима движения информации в окнах — синхронного или асинхронного.

●●● Команда /DG. Параметр G (GОРИЗ) делит экран по горизонтали.

●●● Команда /DW. Параметр W (WЕРТ) делит экран по вертикали.

●●● Команда /DO. Параметр O (ОДНО) отменяет деление экрана на окна.

●●● Команда /DS. Параметр S (SИНХРОН) осуществляет синхронное передвижение (сканирование) информации в окнах. Например, в одном окне выводятся формулы, а в другом окне — значения, вычисленные по этим формулам. При движении указателя вдоль границы раздела будет одновременно движение информации в обоих окнах. Вы сможете одновременно увидеть и формулу, и вычисленное значение для каждой клетки.

●●● Команда /DA. Параметр A (AСИНХР.) позволяет оставить одно окно неподвижным и передвигать информацию только в том окне, где находится указатель. По умолчанию установлено асинхронное передвижение информации.

Рассмотрим пример разделения экрана на два вертикальных окна и установки в них синхронного сканирования (см. табл. 4.10). В левом окне вы предполагаете вывод формул, в правом — значений. Установите указатель в колонку D и введите команду /D. Появится меню.

Для разделения экрана по вертикали введите параметр W. Произойдет разделение экрана на два окна: в левом окне находятся колонки A, B, C, в правом окне — D, E, F, G. Одновременно нажмите левую нижнюю пустую клавишу (**ВПДР**) и клавишу ;, и указатель переместится из правого окна графы D в левое окно в графу C.

Для вывода формул в левом окне воспользуйтесь командой **/AF.** После ее ввода в клетках левого окна появятся формулы. Вновь нажмите клавишу **ВПДР** и клавишу ;. Указатель переместится в правое окно.

Для организации синхронного сканирования в правом окне необходимо иметь идентичные левому окну колонки A, B, C. Сделайте это с помощью указателя клетки (см. табл. 4.4). Нажмите клавиши **=A**. Вы увидите, что в правом окне появились три колонки A, B, C. Не забудьте, что по умолчанию установлено асинхронное сканирование. В связи с этим, прежде чем начать просмотр таблицы, установите режим сканирования, вводя команду разделения по два окна с параметром S, т. е. /**DS**. Теперь, перемещая указатель вдоль границы раздела вниз, вы будете наблюдать синхронное сканирование информации в обоих окнах, что дает возможность одновременно иметь и формулу, и ее значение.

Обратите внимание! При сканировании движение начинается сначала в левом окне, а затем указатель переходит в правое окно и просмотр информации продолжается уже в этом окне. После окончания сканирования фиксация рядов идентична в обоих окнах.

F (**FОРМАТ**). *Изменение формата представления данных в клетках*

Основное назначение этой команды — создание таблицы, удобной для работы: красиво и компактно располо-

жены тексты, числа заданы в одном формате, выровнены все цифровые позиции в соседних клетках. Однако пользоваться этой командой необязательно, если вас устраивают форматы, действующие по умолчанию, а именно: текст в клетках всегда выравнивается по левому краю, а числа — по правому краю, при этом учитывается наиболее рациональное представление этого числа.

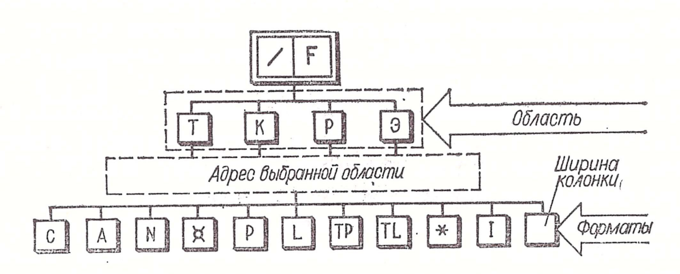


Рис. 4.18. Структура команды изменения формата представления данных F (FОРМАТ)

Формирование фоманды F задается четырьмя уровнями (рис. 4.18). После ввода команды следует определить область установки форматов в таблице: вся таблица (Т), колонка (К), ряд (Р), клетка или группа клеток (Э). Затем вы должны указать конкретный адрес этой области, например, вводя букву колонки или цифру ряда. На последнем уровне вам предлагается задать формат выбранной области:

С (СЕЛ ЧИСЛ) — округление чисел до целого значения;

А (АДАПТИВ) — рациональное представление чисел в клетках;

N (NОРМАЛИЗ) — нормализация чисел;

¤ — точность представления чисел — два знака после запятой;

P (PPAB) — выравнивание чисел по правому краю;

L (LEB) — выравнивание чисел по левому краю;

TP (TEKC PPAB) — выравнивание текста по правому краю;

TL (TEKC LEB) — выравнивание текста по левому краю;

* показывает числа в виде линейной диаграммы;

I (ICXOДH) — восстановление форматов, устанавливаемых по умолчанию: A, P, TL;

ШИР. КОЛОНКИ устанавливает ширину колонки, для чего достаточно ввести число, соответствующее этой ширине.

В табл. 4.10 показано, как представить числа в колонке D с точностью двух знаков после запятой. Второй пример ознакомит вас с формой представления чисел блока в виде линейной диаграммы.

I (IСПОЛНИТЬ). *Исполнение командного файла*

При многократной работе с электронной таблицей вам приходится выполнять ряд однотипных действий. Для автоматизации этих действий можно создать командный файл. Запускает его в работу команда I. Для овладения этим методом работы советуем обратиться к дополнительной литературе [4], где подробно изложены процесс создания командного файла и стратегия работы с ним.

4.4.6. ФУНКЦИИ ВАРИТАБа

При формировании формул вы часто сталкиваетесь с необходимостью вычисления функций. В ВАРИТАБе предусмотрено универсальное средство, которое позволит по указанному имени функции произвести ее вычисление. Вам достаточно только правильно записать и использовать имя функции. В табл. 4.11 приведены условные обозначе-

Таблица 4.11

Функции

Условное обозначение	Математическое обозначение
ПИ	3,14
АБС (X)	$\|X\|$
ЦЕЛ (X)	$[X]$ — целая часть числа
ЕКС (X)	e^X
ЛН (X)	$\ln X$
СИН (X)	$\sin X$
КОС (X)	$\cos X$
ТАН (X)	$\mathrm{tg}\ X$
ККОР (X)	$\sqrt{X}$
ЛОГ 10 (X)	$\lg X$
АСИН (X)	$\arcsin X$
АКОС (X)	$\arccos X$
АТАН (X)	$\mathrm{arctg}\ X$
ИФ	Если
АНД	И
ОР	ИЛИ
НЕТ	Отрицание
СУМ (X_1, ..., X_n)	$X_1 + X_2 + \ldots + X_n$
МИН (X_1, ..., X_n)	Выбор минимального числа
МАК (X_1, ..., X_n)	Выбор максимального числа
СРЕДНЕЕ (X_1, ..., X_n)	$X_1 + X_2 + \ldots + X_n$
НД	Недействительное значение
ОШИБОК	Ошибка
ПОДБОР (ключ, диапазон клеток)	Поиск в двух соседних колонках (рядах) значения по ключу

ПРЦ (норма дисконтирования, диапазон клеток) + первоначальное значение	Финансовая операция. Стоимость приведенная к текущему моменту, с учетом затрат и будущих доходов

ния таких функций. Аргумент *X*, указанный в скобках, может быть числом, арифметическим выражением, ссылкой (имя клетки). Ознакомьтесь с некоторыми функциями.

●●● ЦЕЛ (X). Целая часть числа образуется отбрасыванием дробной части:

ЦЕЛ (12,3)→12 ЦЕЛ (12,8)→12

●●● Для всех тригонометрических функций задается аргумент в радианах.

●●● При вычислении обратных тригонометрических функций результат получается в радианах.

●●● Для функций СУМ, МИН, МАК, СРЕДНЕЕ аргументом, кроме указанных выше, может быть также блок или диапазон клеток. Например, для вычисления суммы значений, содержащихся в клетках A5, A6, A7, A8, B3, B6, D1, D2, E1, E2, используется функция СУМ (A5:A8, B3, B6, D1:E2).

●●● ПОДБОР (ключ, диапазон клеток). Эта функция осуществляет поиск значения по ключу в двух рядом стоящих рядах или колонках. Ключом могут быть число, ссылка, арифметическое выражение. Задаваемое или вычисленное значение ключа сравнивается с числами, находящимися в указанном диапазоне клеток первой колонки (ряда). Если найдено число, равное значению ключа, а при его отсутствии ближайшее к нему, но меньшее по значению, то в клетку, где используется эта фукнция, заносится число из второй колонки (ряда). Причем ряд (колонка) этого числа определяется рядом, где находится число, найденное по ключу.

Например, в пятом ряду (см. с. 196) расположены числа, сравниваемые с ключом, а в шестом ряду — выбираемые значения. В клетке E4 значение определяется как резуль-

тат функции ПОДБОР (C4, A5:E5), где C4 — ключ, значение которого равно 37; A5:E5 — числа, с которыми сравнивается значение ключа C4. В результате сравнения ближайшим числом к числу 37 и не превосходящим его по значению будет 30. Из клетки C6 следующего ряда, находящейся под клеткой C5 с числом 30, будет выбрано число 400 и записано в клетку E4.

	A	B	C	D	E
4			37		400
5	10	20	30	40	50
6	786	350	400	800	300

Ключ

Числа, с которыми сравнивается ключ

Выбираемые значения

Отметим некоторую особенность. При формировании в таблице чисел, сравниваемых со значением ключа, их надо располагать в порядке возрастания.

●●● Функции НД, ОШИБК используются как вспомогательные в логической функции либо появляются в таблице как результат определенных действий.

●●● Функция ПРЦ предназначена для выполнения финансового расчета. Определяется приведенный к текущему моменту суммарный доход с учетом первоначальных затрат и чистых доходов от реализации по проекту. Форма записи:

ПРЦ (норма дисконтирования, диапазон клеток) + + клетка с данными о первоначальных затратах.

Пример. Задайтесь нормой дисконтирования 10 % и запишите ее в виде десятичного числа в клетку A2. Первоначальные затраты на приобретение оборудования составляют 10 000 руб. Это значение надо записать в клетку A4 со знаком —. Ожидаемая прибыль, которая составит в течение трех лет 5000, 10 000, 20 000 руб., занесена в клетки A6, A7, A8. Для определения суммарного дохода, приведен-

ного к текущему моменту с учетом первоначальных затрат, используйте функцию

ПРЦ (А2, А6:А8)+А4

●●● Логические функции ИФ, АНД, ОР, НЕТ обеспечат вам выбор операции по результату сравнения. Общий формат функции, реализующий условный оператор: IF (условие, выражение 1, выражение 2).

Условие — это математические выражения, связанные между собой знаками отношений <, <=, >, >=, =, < > (не равно) и при необходимости — логическими функциями АНД, ОР, НЕТ.

Выражение 1 выполняется, если выполнено условие. Выражение 2 выполняется, если условие не выполнено.

4.5. ПРОВЕРЯЕМ СЕБЯ. КАК УСВОЕН ВАРИТАБ?

Упражнения

4.1. В чем суть процедуры построения электронной таблицы?

4.2. Имеется ли возможность одновременно хранить в оперативной памяти несколько таблиц и организовать между ними связь?

4.3. Какие управляющие символы используются в ВАРИТАБе?

4.4. Понятие «место» в электронной таблице.

4.5. Приведите пример блока.

4.6. Назначение команды ВАРИТАБа и ее структура.

4.7. Как классифицируются команды?

4.8. Какими командами удаляют содержимое клетки или редактируют ее?

4.9. Когда рекомендуется применять команды копирования К и размножения R?

4.10. Как вставить новый ряд (колонку) или переместить ряд (колонку)?

4.11. Какие команды организуют запись таблицы на

диск и ее считывание с диска? Какова стратегия работы с этими командами?

4.12. Как стирается таблица?

Ответы к упражнениям

4.1. Процедура построения электронной таблицы сводится к следующему:

- в виде текста вводите название таблицы;
- формируете шапку таблицы;
- вводите данные и формулы в каждую клетку таблицы;
- формируете итоговые строки;
- печатаете таблицу;
- записываете таблицу на диск.

4.2. В оперативной памяти находится одна электронная таблица. Однако, разделив условно эту таблицу на несколько частей, можно занять их вспомогательными таблицами, необходимыми для решения всей задачи в форме общей электронной таблицы. Имя присваивается всей таблице в целом и в виде файла хранится на диске. Связь между вспомогательными таблицами и основной организуется с помощью ссылок (имен клеток). Так, в параграфе 4.3 показано, как рассчитать заработную плату (основная таблица) с учетом подоходного налога, представленного двумя вспомогательными таблицами.

4.3. Символы / = ! ; ← " ' ⇤ или CTRL Z

Для передвижения указателя и курсора используются клавиши направления либо одновременно следует нажать две клавиши — буквенную и **CTRL**:

CTRL S (←) **CTRL D** (→) **CTRL E** (↑) **CTRL X** (↓)

4.4. «Местом» в электронной таблице служат ряд, колонка, клетка, блок, вся таблица, с которыми производятся действия.

4.5. Блок — это две ссылки, разделенные двоеточием, например: A3:C5. В этот блок включено девять клеток: A3, A4, A5, B3, B4, B5, C3, C4, C5.

4.6. Команда ВАРИТАБа производит действия в электронной таблице в зависимости от используемых параметров. Формирование команды начинается с ввода символа / и первой латинской буквы названия команды. Регистры переключать не надо. Затем в соответствии с предлагаемой подсказкой выбираете и вводите параметры или в виде первой буквы названия параметра, или в виде числа. При этом если после набора команды содержимое экрана не изменилось, то следует нажать клавишу ввода ЕТ. Например, для увеличения ширины колонки А до 35 символов используйте команду форматирования /FKA ЕТ 35 ЕТ

4.7. По своему назначению команды ВАРИТАБа можно представить в виде трех групп:

- команды изменения содержимого таблицы;
- команды взаимодействия таблицы с памятью и принтером;
- команды управления режимами работы и формой представления.

4.8. Команда В удаляет содержимое клетки, а команда Р редактирует.

4.9. Для получения одного экземпляра копирования рекомендуется воспользоваться командой К. Для копирования одного «места» в несколько «мест» воспользуйтесь командой R, учитывая следующее ограничение: нельзя размножать блок, колонку, ряд.

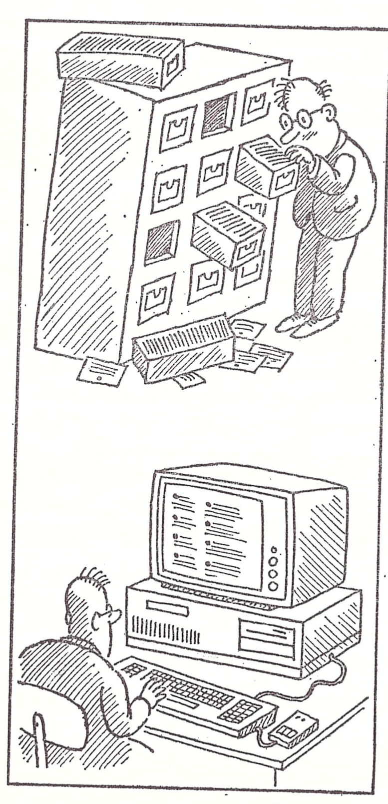
4.10. Команда W вставляет ряд (колонку). Команда М перемещает ряд (колонку).

4.11. Команда S записывает таблицу на диск. Команда Т считывает таблицу с диска. При работе этих команд следует указать имя таблицы. Если вы забыли имя таблицы, то воспользуйтесь предоставляемым в ваше распоряжение меню вызова каталога имен, хранящихся на этом диске файлов.

4.12. Из оперативной памяти — командой N, с диска — командой L.

5

ПРИМЕНЯЕМ КАРТ1715 — СИСТЕМУ ВЕДЕНИЯ КАРТОТЕКИ



5.1. ЗАЧЕМ НУЖЕН КАРТ1715?

Назначение

КАРТ1715 представляет собой пакет прикладных программ, предназначенный для создания и ведения автоматизированных картотек различного назначения. Примерами таких картотек могут служить деловые календари и записные книжки, картотеки отделов кадров учреждений, библиотечные картотеки, картотеки складских хозяйств и т. п.

Структура данных

По существу пакет КАРТ1715 является диалоговой информационно-справочной системой, в которой информация может храниться в виде совокупности карточек. Каждая такая карточка может включать до

26 информационных полей. В каждое поле заносится текстовая (символьная) информация, представляющая собой, например, фамилию, имя и отчество работника в картотеке отдела кадров, название книги в библиотечной картотеке или тематику конференции в деловом календаре. Число символов в информационном поле не может превышать 1404. Суммарная длина полей в одной карточке ограничена также 1404 символами. Количество карточек в картотеке может достигать 30 тыс. При этом, конечно, следует учитывать ограничение рабочего пространства на используемом гибком магнитном диске. Объем пакета КАРТ1715 составляет 56 Кбайт.

Основные функции

Основные функции, реализуемые КАРТ1715, сводятся к следующему:

- формирование картотеки, связанное с описанием структуры карточки и занесением данных;
- коррекция как структуры карточки, так и данных на отдельных ее полях;
- ведение картотеки, состоящее в добавлении новых и удалении имеющихся карточек;
- последовательный просмотр карточек;
- получение простой справки в виде сведений, помещенных в какую-либо карточку, по так называемому ключевому слову, в качестве которого могут быть указаны, например, фамилия или табельный номер работника;
- получение более сложной справки в виде выборки карточек (или отдельных полей) из их исходной совокупности, например данных о работниках, имеющих высшее образование;
- просмотр полученной справочной информации на экране дисплея, запись ее на магнитный диск или вывод на печатающее устройство.

Особенности КАРТ1715

Сравнивая пакет КАРТ1715 с известными системами управления базами данных, следует указать на простоту диалога, который ведется пользователем при работе с данным пакетом. Простота управления пакетом объясняется хорошей наглядностью информации, выводимой на экран дисплея, достаточным уровнем подсказок со стороны программ пакета и ограниченным набором команд, которые вы должны освоить для реализации разнообразных функций КАРТ1715. В связи с этим пакет может применяться лицами, не имеющими специальной подготовки в области использования ЭВМ.

Существенным ограничением в использовании пакета КАРТ1715 является то, что заносимая в картотеку информация может быть только символьного (нечислового) вида. Поиск в картотеке возможен лишь на совпадение с задаваемым ключевым словом. Поэтому с помощью пакета нельзя производить поиск по критерию «больше» или «меньше» вводимого порогового значения. Например, невозможна реализация запроса в картотеке отдела кадров для вывода списка работников моложе тридцати лет и т. п. В пакете не предусмотрены такие типовые для работы с базами данных операции, как сортировка и арифметические действия над данными. Например, нет возможности подсчитать средний возраст работников учреждения или суммарную стоимость товаров на складе.

И все же следует ожидать, что пакет прикладных программ КАРТ1715, благодаря простоте его освоения и использования, найдет широкое применение в различных системах хранения и обработки нечисловой информации.

Ниже на простом примере ведения картотеки домашней библиотеки демонстрируется выполнение основных операций по управлению пакетом КАРТ1715. Предлагаем вам проверить на компьютере приведенные последовательности команд, вводимых в диалоге с программами

пакета. Следуйте нашим советам, и вы убедитесь в простоте и эффективности средств КАРТ1715.

5.2. ЗНАКОМСТВО С КАРТ1715

Определите цель вашей работы

Итак, вы намереваетесь составить картотеку вашей домашней библиотеки. Вначале уясните план ваших действий. В первую очередь вам следует выбрать и зафиксировать с помощью КАРТ1715 структуру карточки с данными об отдельной книге. Затем можно будет приступить к заполнению картотеки учетной информацией. Наибольший интерес у вас должна вызвать работа с картотекой, в процессе которой вы сможете формировать различные запросы и получать необходимые справки.

При знакомстве со средствами пакета программ КАРТ1715 рекомендуем вам в первую очередь просмотреть текстовой файл, содержащий краткую информацию о назначении пакета, его основных функциях и перечне команд. Для этого выполните ряд указанных ниже действий.

Загрузите операционную систему

Включите питание компьютера. Установите гибкий магнитный диск с операционной системой и пакетом программ КАРТ1715 на дисковод А. Закройте защелку дископриемника, после чего выполнится загрузка операционной системы.

Ознакомьтесь с меню КАРТ1715

Установите режим ввода букв русского алфавита, переключив клавишу **SI/SO** в положение, в котором загорится расположенный рядом индикатор. Нажмите клавишу пробела. Введите команду: **a>**карт1715 **ЕТ**. По окончании загрузки пакета на экране появится соответствующая заставка, а через несколько секунд — основное меню, вид которого приведен ниже.

```
 КАРТИН В-1                                     +
                                                +
ГЛАВНЫЕ ФУНКЦИИ :                               +  ВТОРИЧНЫЕ ФУНКЦИИ :
                                                +
  ==>  ДАННЫЕ                                   +    ==>  ПОИСК
       ЗАДАНИЕ ФОРМАТА                          +         АНАЛИЗ
       СЕРВИС ОПЕРАЦИОННОЙ СИСТ.                +         СОЗДАН.
                                                +         ВОССТ.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ГЛАВНАЯ ФУНКЦИЯ =   ДАННЫЕ
ВТОРИЧНАЯ ФУНКЦИЯ = ПОИСК

ФАЙЛ =*
СМЕНА ДИСКА =  НЕТ

ДЛЯ ВЫБОРА НАЖАТЬ КЛ. (Г,В,Ф,Д) ИЛИ КЛ. ЕСЦ:
```

Информация меню, выводимая в верхней половине экрана, представляет собой перечень доступных для пользователя функций: главных и вторичных.

●●● Под главными функциями пакета понимаются следующие:

◆ работа с данными, вводимыми или хранящимися в картотеке (позиция меню «Данные»);

◆ задание или коррекция структуры карточки («Задание формата»);

◆ копирование файлов картотеки и удаление прочих файлов («Сервис операционной системы»).

●●● Вторичная функция конкретизирует то действие, которое может быть выполнено в данный момент. Убедитесь, нажимая клавишу **Г**, что перечень вторичных функций меняется в зависимости от положения стрелки—указателя главных функций. Заметьте также, что по достижении нижней строки меню главных функций стрелка будет переброшена вновь в начало меню, если в очередной раз нажать клавишу **Г.** Аналогичным образом вы можете

управлять перемещением стрелки—указателя вторичных функций, нажимая клавишу **В**.

Обратите внимание! В нижней половине экрана выводятся наименования тех функций, которые установлены в данный момент.

Сделайте вызов справочной информации

В целях просмотра текстового файла со справочной информацией о пакете КАРТ1715 установите функции: главную — «Данные», вторичную — «Поиск». Нажмите клавишу **Ф** для последующего ввода имени файла справочной информации. Обратите внимание на то, что справа от сообщения ″ФАЙЛ=″ появилось многоточие, число точек в котором соответствует допустимому числу символов в вводимом имени. Введите имя **курс**, завершив ввод нажатием клавиш **ЕТ** и **ESC**. Проанализируйте появившееся в нижней части экрана меню:

ДЛЯ СМЕНЫ — НАЖАТЬ КЛ. ″С″,
ДЛЯ ВЫПОЛНЕНИЯ — КЛ. ″В″ ИЛИ ″К″ —
КОНЕЦ РАБОТЫ

Однобуквенные команды в этом меню означают следующее:

С — возврат к основному меню для смены функций;
В — запуск выполнения установленных функций;
К — выход из пакета КАРТ1715.

Ознакомьтесь со справочной информацией

Нажмите клавишу **В**. Через несколько секунд на экране появится первая страница (карточка) поясняющего текста. В нижней же части экрана появится новое меню с перечнем команд, позволяющих просматривать файл справочной информации. Нижеприведенные однобуквенные команды вводите при нажатой клавише **CTRL**, что в дальнейшем будет помечаться символом кавычки ″ слева от буквы:

″**Р** — переход на первую страницу (карточку) файла;
″**Ц** — переход на последнюю страницу файла;

"А — переход к предыдущей странице;
"Ф — переход к следующей странице.

Используя данные команды, просмотрите текст, поясняющий работу пакета КАРТ1715. Для возвращения в основное меню нажмите клавиши **М**, **Е** и **ЕТ**. На этом первое знакомство с КАРТ1715 заканчивается.

5.3. ФОРМИРУЕМ СТРУКТУРУ КАРТОЧКИ

Вы уже поставили себе цель — создать картотеку домашней библиотеки. Для ее размещения может быть использован диск с пакетом КАРТ1715, на котором имеется достаточно свободного пространства. Все последующие действия старайтесь выполнить так, как будет рекомендовано ниже.

Создайте файл формата карточки

Находясь в основном меню КАРТ1715, установите следующие функции: главную — «Задание формата», вторичную — «Создание». Нажмите клавишу **Ф** и введите имя создаваемой картотеки, например **библи**. Далее нажмите клавишу **ESC**, а затем — **В**. В результате будет открыт файл формата карточки **библи.фмт**, определением которого вы сейчас и займетесь.

Выберите функцию создания формы карточки

Обратите внимание на сообщения, появившиеся на экране. В верхней его части выведено сообщение «ВЫБОР ФУНКЦИИ», а в нижней — меню функций, которые можно выполнить при определении формата карточки:

Э — создание или коррекция формы карточки с возможностью разметки всех ее полей (**Э** вводится в верхнем регистре);
П — определение формата избранного поля;
У — удаление поля из формы карточки;
Ф — задание основного формата печати карточек;

Д — запись созданного или скорректированного файла формата на диск;
М — возврат в основное меню.

Введите команду **Э** при нажатой клавише верхнего регистра (**ВПДР**).

Ознакомьтесь с меню режима коррекции экрана

На экране — очередное меню, соответствующее состоянию «КОРРЕКТ. ЭКРАНА» (в нашем случае — создания экранной формы карточки):

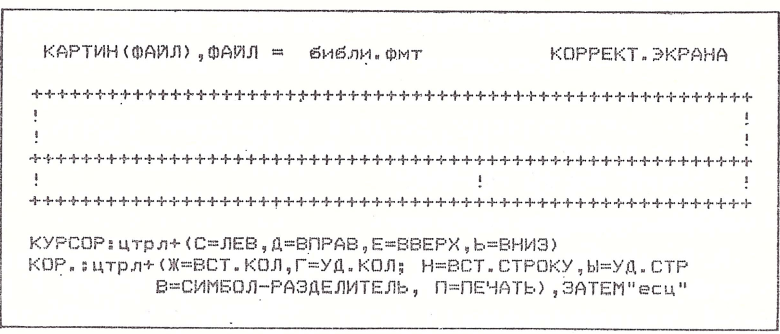
"С (или ←) — перемещение курсора влево на одну позицию;
"Д (или →) — перемещение курсора вправо на одну позицию;
"Е (или ↑) — перемещение курсора вверх на одну строку;
"Ь (или ↓) — перемещение курсора вниз на одну строку;
"Ж, **"Г** — соответственно вставка и удаление колонки;
"Н, **"Ы** — соответственно вставка и удаление строки;
"В — ввод символа—разделителя полей карточки;
"П — печать формы карточки.

Создайте структуру карточки

Для определения структуры карточки вам необходимо предварительно решить, какие данные, какой длины и в какой последовательности будут храниться в карточке. Допустим, вы ограничились такими сведениями о книге в вашей домашней библиотеке: сведения об авторстве, название книги и год издания и решили расположить эти данные в полях, как показано ниже.

```
------------------------------------------------------------------------
!Название:                                                             !
!                                                                      !
------------------------------------------------------------------------
!Автор:                                    !Год изд.                   !
------------------------------------------------------------------------
```

В состоянии «КОРРЕКТ.ЭКРАНА» начинайте разметку полей карточки. С этой целью переместите курсор в крайнюю левую верхнюю позицию рабочей области экрана, после чего с помощью команды **″В** проведите горизонтальную линию необходимой длины. Далее сместите курсор на три строки вниз, верните его в крайнюю левую позицию экрана и проведите вторую линию той же длины. Таким же образом проведите на экране и третью горизонтальную линию. Перемещая курсор соответствующим образом, проведите вертикальные линии с помощью символа «!». В результате будет получена экранная форма карточки, которую при желании вы можете распечатать с помощью команды **″П**. Результат такой печати представлен ниже.



```
 КАРТИН(ФАЙЛ),ФАЙЛ =  библи.фмт           КОРРЕКТ.ЭКРАНА

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!                                                        !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!                                    !                   !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

КУРСОР:цтрл+(С=ЛЕВ,Д=ВПРАВ,Е=ВВЕРХ,Ь=ВНИЗ)
КОР.:цтрл+(Ж=ВСТ.КОЛ,Г=УД.КОЛ; Н=ВСТ.СТРОКУ,Ы=УД.СТР
          В=СИМБОЛ-РАЗДЕЛИТЕЛЬ, П=ПЕЧАТЬ),ЗАТЕМ"есц"
```

Завершив разработку формы карточки, нажмите клавишу **ESC**. В результате вы вернетесь в меню «ВЫБОР ФУНКЦИИ».

Ознакомьтесь с меню определения полей карточки

Введите команду **П**, после чего осуществится переход в состояние «ОПРЕДЕЛ.ПОЛЯ». Ознакомьтесь с новым меню, выведенным в нижней части экрана:

Н — задание символа-идентификатора, которым заполнится поле по окончании его создания, а также двухбуквенного наименования для последующих ссылок на это поле;
И — задание режима индексирования слов поля, т. е. пометки тех слов, которые будут просматриваться при выполнении поиска в картотеке на совпадение с ключевым словом;
З — введение текста заголовка, поясняющего хранящиеся в поле данные;
С, **К** — задание позиции соответственно левого верхнего и правого нижнего углов создаваемого поля.

Определите поля карточки

Введите команду **Н** и в ответ на соответствующий запрос введите символ-идентификатор (например, А) и сразу же после него — двухбуквенное наименование поля, например **НА**. Далее введите с помощью команды З текст заголовка поля, например **Название:**. Нажимая клавишу **И**, установите режим индексации слов **ВСЕ**, что соответствует режиму индексации всех слов в тексте поля. Затем переместите курсор в левый верхний угол первого поля карточки на экране и нажмите клавишу **С**, зафиксировав таким образом начало поля. Переместите курсор в правый нижний угол поля и с помощью команды **К** зафиксируйте конец создаваемого поля. На экране появится созданное поле с заголовком и символом-идентификатором.

В конце нажмите клавишу **ESC**.

Аналогичным образом определите второе поле, задав идентификатор **Б**, наименование поля **АВ** и введя заголовок **Автор:**. При создании третьего поля задайте идентификатор **Г**, наименование поля **ГО** и заголовок **Год изд.:**. В результате на экране будет создана структура карточки.

```
  КАРТИН(ФАЙЛ),ФАЙЛ =  библи.фмт              ВЫБОР ФУНКЦИИ

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: аааааааааааааааааааааааааааааааааааааааааа !
! ааааааааааааааааааааааааааааааааааааааааааааааааааа !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: бббббббббббббббббббббббб !Год изд. ггггггггг !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


 Э=КОР.ЭКРАНА, П=КОР./СОЗДАНИЕ ПОЛЯ, У=УДАЛИТЬ ПОЛЕ,
 Ф=ФОРМАТ ПЕЧАТИ
 ВЫХОД: "Д"=Ф-Л НА ДИСК, В МЕНЮ -"М"
 УКАЖИ КОД ФУНКЦИИ:
```

Задайте формат печати карточки

Завершите определение структуры карточки заданием формата распечатки картотеки. Для этого введите в состоянии «ВЫБОР ФУНКЦИИ» команду **Ф** и после появления на экране параметров формата печати с помощью команды **З** измените число записей на странице, введя цифру **5**. Затем нажмите клавишу **ESC** и в конце — **Д** для записи созданного файла на диск.

5.4. ЗАПОЛНЯЕМ КАРТОТЕКУ ДАННЫМИ

Допустим, список книг в вашей домашней библиотеке начинается следующим перечнем:

1. Пушкин А. С. Собрание сочинений в 10 томах. 1959—1962.
2. Мейлах Б. Жизнь Александра Пушкина. 1974.
3. Пушкин А. С. Стихотворения и поэмы. 1976.
4. Новиков И. А. Пушкин в Михайловском. 1982.
5. Пушкин А. С. Сочинения в 3 томах. 1985—1986.

Теперь занесите эти данные в картотеку, формат которой уже вами определен, соблюдая указанный ниже порядок работы.

Определите файл данных

В основном меню установите функции: главную — «Данные», вторичную — «Создание». Введите команду **Ф** и укажите имя файла данных **библи.**

Вызовите меню создания данных

Нажмите клавишу **ESC**, а затем введите команду **В** (выполнить). В нижней области экрана появится меню, из которого вам пока потребуются команды:

- ДОБАВИТЬ — включение в картотеку новых карточек при создании или расширении картотеки;
- КОРРЕКТ — коррекция содержимого изображенной на экране карточки;
- УДАЛИТЬ — удаление изображенной карточки при нажатии клавиши **ЕТ**;
- ПЕЧАТЬ — запись выборки карточек на диск или вывод на печатающее устройство;
- СОХРАН — запись введенных карточек в картотеку на диске;
- МЕНЮ — возврат к основному меню.

Все перечисленные команды вводятся двумя первыми буквами с последующим автоматическим расширением слова команды.

Введите данные и проконтролируйте их ввод

Введите команду ДО, после чего на экране появится структура карточки и курсор переместится в начало первого поля карточки, в которое введите текст **Собрание сочинений в 10 томах.** Закончив ввод, нажмите клавишу **ЕТ.** Произойдет переход к очередному полю, в которое введите **Пушкин А. С.** В третье поле занесите **1959—1962.** В конце нажмите клавишу **ESC,** а затем введите команду **С** для записи карточки в буфер оперативной памяти. При этом осуществится переход к следующей карточке.

Аналогичным образом заполните остальные карточки.

Для контроля результатов ввода воспользуйтесь командами просмотра картотеки, приведенными в параграфе 5.2. При необходимости распечатки сформированной картотеки введите команду **ПЕ**. Результат печати первой страницы с пятью первыми карточками приведен ниже.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Собрание сочинений в 10 томах                 !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                 !Год изд. 1959-1962  !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Жизнь Александра Пушкина                      !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Мейлах Б.                   !Год изд. 1974       !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Стихотворения и поэмы                         !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                 !Год изд. 1976       !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Пушкин в Михайловском                         !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Новиков И.А.                !Год изд. 1982       !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Сочинения в 3 томах                           !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                 !Год изд. 1985-1986  !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

По мере необходимости проведите коррекцию карточек

При обнаружении ошибок в какой-либо карточке, используя команды просмотра картотеки, выведите содер-

жание этой карточки на экран. Затем введите команду **КО** и проведите коррекцию указываемого дополнительно поля.

Для удаления карточки целиком выведите ее на экран, введите команду **УД** и нажмите клавишу **ET**.

5.5. ПОЛУЧАЕМ СПРАВКУ ИЗ КАРТОТЕКИ

Для получения из картотеки той или иной справочной информации придерживайтесь наших рекомендаций и нижеследующего порядка действий.

Установите режим поиска

Установите функции основного меню: главную — «Данные», вторичную — «Поиск». Введя команду **Ф**, задайте имя файла вашей картотеки — **библи.** После нажатия клавиши **ESC** введите команду **В** (выполнить). В результате на экран будут выведены первая карточка, хранящаяся в картотеке, служебная информация в верхней части экрана и меню команд в нижней его части, как это показано ниже.

```
 КАРТИН(ПОИСК),ФАЙЛ = а:библи.фил      ВЫВОД
УРОВ.  0 : ЗАПИСЬ 1 ИЗ 5
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Собрание сочинений в 10 томах                 !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                   !Год изд. 1959-1962 !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


ВВОД КОМАНДЫ :
 МАСКА;ВЫБОР;ВКЛЮЧАЯ;КРОМЕ;ИСТОРИЯ;ВОЗВРАТ;СБРОС;СПИСОК;
 ДОБАВИТЬ;ДУБЛЬ;КОРРЕКТ;УДАЛИТЬ;ВВОД;ПИСАТЬ;ФОРМАТ;ПЕЧАТЬ;
 СОХРАН;МЕНЮ
ТЕКСТ:цтрл+(Р=1ЗАП,Ц=ПОСЛ,А=ВОЗВ,Ф=ВПЕР;
      КОМ:Ь=УДАЛИТЬ,Х=ВОЗВ.НА СИМ)
```

Обратите внимание! Служебная информация включает имя файла данных картотеки **библи.фил**, число карточек в ней и порядковый номер той карточки, которая отображается в данный момент на экране в виде сообщения "Запись 1 из 5". "Уровень 0" соответствует исходной выборке, в которую включены все хранящиеся в картотеке карточки.

Просмотрите вашу картотеку

Рекомендуем вам просмотреть хранящуюся на гибком магнитном диске картотеку. При ограниченном ее объеме «перелистывание» карточек — наиболее простой способ получения нужных сведений. Для просмотра картотеки пользуйтесь следующими командами из отображаемого на экране меню:

"**Ф** — переход к следующей карточке;
"**А** — возврат к предыдущей карточке;
"**Р** — переход к первой карточке;
"**Ц** — переход к последней карточке.

Распечатайте содержимое картотеки

При желании вы можете распечатать содержимое картотеки вашей домашней библиотеки. При этом будет использован формат печати, который вы задали по окончании определения структуры карточки еще до ввода данных в картотеку. Вспомните, единственное отклонение от стандартного формата печати состояло в размещении пяти карточек на одной странице. Введите команду **ПЕ** (печать) и после появления на экране меню, предлагающего вам уточнить режим печати, нажмите клавишу **ESC**, а затем введите команду **В** (выполнить). В результате вы получите твердую копию содержимого картотеки вашей домашней библиотеки. Первая страница распечатки соответствует приведенной на стр. 212.

В чем суть режима поиска?

Пакет КАРТ1715 предоставляет вам возможность поиска данных в картотеке. Поиск производится по образцу

(ключевому слову), вводимому пользователем. В основном режиме поиска с образцом сравниваются лишь индексированные слова. В вашем случае при определении структуры карточки вы установили режим, в котором все вводимые в картотеку слова подвергались индексированию, т. е. все слова будут просматриваться при поиске.

Поиск вы можете осуществлять как локальный в рамках одного из полей карточек, так и глобальный по всем без исключения полям.

Для последующего выбора ключевых слов поиска уясните те полезные эффекты, которые могут быть получены при использовании специальных символов ? и +. Символ ? соответствует любому символу в указанной позиции ключевого слова. Символ +, стоящий в начале ключевого слова, допускает любую последовательность символов, примыкающую к слову слева, а символ + в конце ключевого слова — произвольное его продолжение.

Уясните еще одно важное понятие — уровень выборки. Исходная совокупность карточек является выборкой уровня 0. В результате выполнения первой операции поиска будет сформирована выборка уровня 1, в которую войдут карточки с данными, удовлетворяющими заданному критерию поиска. Текущая выборка может быть вновь подвергнута операции поиска с получением выборки уровня 2 и т. д. КАРТ1715 позволяет достигать при поиске уровня 99.

Содержание текущей выборки по вашему желанию может быть просмотрено на экране, выведено на печать или записано в новый файл данных.

Ознакомьтесь с основными командами поиска

●●● Для начала рекомендуем вам освоить три основные команды поиска:

◆ ВЫБОР — из текущей выборки формируется новая выборка, включающая карточки, в которых обнаружено совпадение с введенным ключевым словом;

◆ КРОМЕ — из текущей выборки удаляются карточки, в которых обнаружено совпадение с ключевым словом;

◆ ВКЛЮЧАЯ — к текущей выборке добавляются новые карточки из исходной выборки, в которых обнаружено совпадение с ключевым словом.

При выполнении этих трех команд согласно подсказкам КАРТ1715 вам вначале надо будет ввести двухбуквенное наименование поля, в котором будет производиться поиск, затем — символ-разделитель "/" и в конце — ключевое слово. Если вы намереваетесь провести глобальный поиск (по всем полям карточек), то наименование поля опустите.

●●● Те же данные вам следует ввести и при реализации еще одной полезной команды — СПИСОК, в результате выполнения которой на экран выводится список слов, соответствующих заданному ключевому слову.

●●● Полезными для вас могут также оказаться и такие вспомогательные команды:

◆ ИСТОРИЯ — на экран выводятся список выполненных команд поиска, наименования использованных при этом полей карточки и ключевых слов, а также число карточек на каждом уровне выборки;

◆ ВОЗВРАТ обеспечивает возврат к предыдущей выборке;

◆ СБРОС возвращает к исходной выборке уровня 0.

Все перечисленные выше команды вводятся первыми двумя буквами их наименований. КАРТ1715 распознает введенную команду и автоматически расширяет ее наименование вправо.

Произведите поиск в картотеке

Допустим, вы желаете получить список названий книг, автором которых является Александр Сергеевич Пушкин. Если пакет КАРТ1715 находится в состоянии «Поиск», то введите команду **ВЫ**. Далее введите двухбуквенное наименование поля **ав**, предназначенного для хранения све-

дений об авторстве издания, затем — символ "/", после чего — ключевое слово **Пушкин**. После нажатия клавиши **ЕТ** на экран будет выведена первая карточка из полученной выборки, а в верхней служебной области экрана вы обнаружите сообщение: «Уров.1: запись 1 из 3». Цифра 3 указывает число карточек, вошедших в выборку, сформированную в результате первой операции поиска.

Распечатайте карточки из полученной выборки, выполнив команду **ПЕ**, как это пояснялось выше. В результате вы будете иметь нижеследующий документ.

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Собрание сочинений в 10 томах                !
!                                                       !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                  !Год изд. 1959-1962 !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Стихотворения и поэмы                        !
!                                                       !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                  !Год изд. 1976      !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Сочинения в 3 томах                          !
!                                                       !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                  !Год изд. 1985-1986 !
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Сравните его с исходной выборкой картотеки, приведенной на стр. 212.

Продолжайте поиск, поставив перед собой цель выявить из выборки те карточки, в которых год издания книг относится к 1980-м годам. Для этого вновь введите команду **ВЫ** (ВЫБОР), а затем **го/198?** ↵. В выборку уровня 2 войдет лишь одна карточка.

```
 КАРТИН(ПОИСК),ФАЙЛ = а:библи.фил      ВЫВОД
УРОВ.  2 : ЗАПИСЬ 1 ИЗ 1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Сочинения в 3 томах                             !
!                                                          !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                  !Год изд. 1985-1986   !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Теперь попробуйте расширить текущую выборку, включив в нее карточки, содержащие наименования книг, посвященных А. С. Пушкину. С этой целью введите команду **ВК** (ВКЛЮЧАЯ) и затем **на/Пушкин┼**. Ниже приведена полученная в результате такого трехступенчатого поиска выборка уровня 3.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Жизнь Александра Пушкина                        !
!                                                          !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Мейлах Б.                    !Год изд. 1974        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Пушкин в Михайловском                           !
!                                                          !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Новиков И.А.                 !Год изд. 1982        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Сочинения в 3 томах                             !
!                                                          !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Пушкин А.С.                  !Год изд. 1985-1986   !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Если вы захотите оставить в этой выборке только те карточки, в которых год издания предшествует 1980-му, то введите команду **КР** (КРОМЕ) с продолжением **го/198?┼**. В результате будет сформирована выборка уровня 4, состоящая лишь из одной карточки.

```
 КАРТИН(ПОИСК),ФАЙЛ = а:библи.фил      ВЫВОД
УРОВ.  4 : ЗАПИСЬ 1 ИЗ 1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Название: Жизнь Александра Пушкина                      !
!                                                        !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!Автор: Мейлах Б.                     !Год изд. 1974     !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Рекомендуем вам ввести команду **ИС** и проанализировать выведеную на экран «историю» операций поиска.

```
 КАРТИН(ПОИСК),ФАЙЛ = а:библи.фил      ВЫВОД
НОВЫЙ УРОВ.   4. ФАЙЛ СОДЕРЖИТ  5 ЗАПИСЕЙ.

УРОВ.  1 : выБОР  ав/пушкин - 3 НАЙДЕНО ЗАПИСЕЙ
УРОВ.  2 : выБОР  го/198?+ - 1 НАЙДЕНО ЗАПИСЕЙ
УРОВ.  3 : вКЛЮЧАЯ на/пушкин+ - 3 НАЙДЕНО ЗАПИСЕЙ
УРОВ.  4 : крОМЕ   го/198?+ - 1 НАЙДЕНО ЗАПИСЕЙ
```

В завершение введите команду **СБ** и вернитесь к исходной выборке уровня 0.

Для освоения команды СПИСОК введите сначала **СП**, а затем **ав/???+**. В результате на экран будет выведен список фамилий авторов книг вашей домашней библиотеки.

```
 КАРТИН(ПОИСК),ФАЙЛ = а:библи.фил      ВЫВОД
    ==========   ИНДЕКС:  ==========
мейлах
новиков
пушкин
```

5.6. ОБЩАЯ СПРАВКА О КАРТ1715

Материалом этого параграфа вы будете пользоваться как справочным. В табл. 5.1 сведены основные команды пакета прикладных программ КАРТ1715, причем сгруппированы они в зависимости от выбранного меню.

Таблица 5.1

Команды управления КАРТ1715

Клавиша	Назначение команды
	Основное меню главных и вторичных функций
Г	Смена главной функции. На экране отображается смещением стрелки-указателя функции в очередную позицию меню главных функций
В	Смена вторичной функции. Отображается смещением стрелки-указателя в очередную позицию меню вторичных функций
Ф	Инициализация запроса ввода имени входного файла. Ввод имени завершается нажатием клавиши ЕТ
Ы	Инициализация запроса имени выходного файла при реализации функции копирования файла
Д	Смена дисковода перед последующей операцией чтения или записи файла на гибкий магнитный диск
ESC	Переход в промежуточное меню: С — сменить функцию; В — выполнить; К — конец работы
	Промежуточное меню при выходе из основного меню
С	Возврат к основному меню для повторения или смены функций
В	Выполнение функций, определенных в основном меню
К	Конец работы в пакете КАРТ1715 с выходом в операционную систему
	Предварительное меню создания (коррекции) файла формата карточки
Э	Переход в меню создания (коррекции) экранной формы карточки. Буква Э вводится в верхнем регистре

Клавиша	Назначение команды
П	Переход в меню создания (коррекции) структуры поля карточки. При создании нового поля КАРТ1715 сразу же запрашивает однобуквенный идентификатор и двухбуквенное имя поля. При коррекции структуры поля запрашивается только его идентификатор
У	Удаление поля в структуре карточки. Запрашивается идентификатор удаляемого поля
Ф	Переход в меню определения основного формата печати карточек
Д	Запись созданного (скорректированного) файла формата карточки на гибкий магнитный диск
М	Возврат в основное меню без изменения файла формата
Меню создания (коррекции) экранной формы карточки	
"Х, "С, ←	Смещение курсора влево на один символ. При использовании функциональной клавиши ← возможна автогенерация ввода
"Л, "Д, →	Смещение курсора вправо на один символ
"К, "Е, ↑	Смещение курсора вверх на одну строку
"Й, "Ь, ↓	Смещение курсора вниз на одну строку
"В	Ввод символа — разделителя полей карточки по горизонтали. Последовательный ввод команды приводит к появлению на экране: *, < >, <—>, <— —> и т. д. В последующем при распечатке карточек предоставляется возможность выбора в качестве символа-разделителя +, *, — или пробела
"Ж	Вставка дополнительной колонки в поле
"Г	Удаление одной колонки из поля
"Н	Вставка дополнительной строки в поле
"Ы	Удаление одной строки из поля
"П	Печать экранной формы карточки

Клавиша	Назначение команды
ESC	Возврат в предварительное меню создания (коррекции) файла формата карточки
	Меню создания (коррекции) структуры поля карточки
З	Инициализация запроса ввода текста заголовка, помещаемого в начало поля. В ответ на запрос вводится заголовок, поясняющий назначение данного поля
И	Задание режима индексации слов поля. При нажатии клавиши **И** чередуются режимы: НЕТ — все слова поля не индексируются; ВЫБ — при вводе или редактировании текста поля предоставляется возможность выборочно индексировать отдельные слова с помощью команды И (при нахождении курсора в выбранном слове); при переходе к очередному слову индексация автоматически сбрасывается; АВТО — при вводе (редактировании) поля предоставляется возможность снимать индексацию слов с помощью команды И, которая первоначально устанавливается автоматически; ВСЕ — все слова поля автоматически индексируются, возможности снятия индексации не предоставляется
С	Фиксация начала поля (ряда и колонки на экране). Предварительно курсор устанавливается в левый верхний угол поля
К	Фиксация конца поля. Предварительно курсор устанавливается в правый нижний угол прямоугольника поля. После фиксации начала и конца поля оно автоматически заполняется ранее выбранным идентификатором с помещением в начало поля введенного заголовка
Н	Замена двухбуквенного имени поля
ESC	Возврат в предварительное меню создания (коррекции) файла формата карточки

Клавиша	Назначение команды
Меню определения основного формата печати карточек	
Р	Задание числа строк на странице
С	Определение режима перехода к печати следующей страницы
В	Задание числа пустых строк от верхнего края листа до текста
Л	Задание числа отступов от левого края листа
З	Задание числа карточек, распечатываемых на одной странице
П	Задание числа строк — пропусков между соседними карточками
ESC	Переход в предварительное меню создания (коррекции) файла формата карточки
Объединенное меню функций создания (коррекции) и поиска данных в картотеке	
"Р	Переход к первой карточке картотеки
"Ц	Переход к последней карточке картотеки
"А	Возврат к предыдущей карточке
"Ф	Переход к следующей карточке
"Ь	Удаление набранной команды
"Х, ←	Смещение курсора в набранной команде на один символ лево
ДО	Добавить. Ввод текста в поля новой карточки, добавляемой к картотеке. При вводе и редактировании текста поля используется локальное меню, выводимое в нижней части экрана. Переход к очередному полю обеспечивается нажатием клавиши ЕТ. Ввод завершается нажатием клавиши ESC с переходом в промежуточное меню: С — сохранить данную карточку в буфере оперативной памяти; М — сбросить ввод; Р — вернуться к редактированию карточки

Клавиша	Назначение команды
УД	Удалить. Удаление из картотеки текущей карточки
ДУ	Дубль. Дублирование текущей карточки с целью ее последующего редактирования
КО	Корректировать. Коррекция текста в текущей карточке
СО	Сохранить. Накопленные в буфере оперативной памяти компьютера изменения к картотеке записываются на гибкий магнитный диск
ВЫ	Выбор. Из текущей выборки карточек образуется новая выборка, включающая только те карточки, в которых обнаружено совпадение с заданным ключевым словом при просмотре индексированных слов указанного поля или всех полей карточек
КР	Кроме. Из текущей выборки карточек удаляются те из них, в которых обнаружено совпадение с ключевым словом
ВК	Включая. В текущую выборку включаются карточки из исходной (уровня 0) выборки, в которых обнаружено совпадение с ключевым словом
ВО	Возврат. Возврат к предыдущей выборке со сбросом текущей
СБ	Сброс. Возврат к исходной выборке (уровня 0) с потерей всех выборок более высокого уровня
ИС	История. Вывод на экран введенных команд поиска и числа найденных при этом карточек
СП	Список. Вывод на экран списка индексированных слов, совпавших с заданным ключевым словом
ВВ	Ввод. Ввод в картотеку данных из файла на гибком магнитном диске
ПИ	Писать. Запись текущей выборки в файл на диске

Клавиша	Назначение команды
МА	Маска. Поиск данных повышенной сложности, запускаемый вводом подкоманд Выбор или Кроме. В отличие от обычного режима поиска просматриваются все, в том числе и неиндексированные, слова полей карточек
ПЕ	Печать. Распечатка или запись на диск карточек текущей выборки. С помощью локального меню можно скорректировать режим печати (записи), задать имя выходного файла
ФО	Формат. Предоставляется возможность замены основного формата печати посредством указания имени другого заранее подготовленного файла формата
МЕ	Меню. Возврат к основному меню КАРТ1715

5.7. ПРОВЕРЯЕМ СЕБЯ. КАК УСВОЕН КАРТ1715?

Упражнения

5.1. Как при коррекции формата карточки расширить ее поле на несколько позиций?

5.2. Каким образом можно вставить в текст какого-либо поля карточки пропущенное при вводе слово?

5.3. Выведите на экран и печать последнюю карточку из вашей картотеки.

5.4. Как просмотреть содержание карточки из картотеки большого объема, если вам неизвестно, в какой части картотеки находится искомая карточка?

5.5. Можно ли при поиске данных в картотеке обеспечить просмотр неиндексируемых слов?

5.6. Распечатайте содержимое картотеки по одному избранному вами полю карточки.

5.7. Удалите вашу картотеку **библи** с магнитного диска.

5.1. В состоянии «КОРРЕКТ.ЭКРАНА» переместите курсор в выбранное вами поле и несколько раз подряд введите команду ″**Ж**.

5.2. В процессе редактирования содержимого карточки переместите курсор в позицию, начиная с которой вы намереваетесь вставить пропущенное слово. Затем выполните раздвижку текста поля посредством нажатия клавиши ″**Ж**. Далее введите в зарезервированное место вставляемое слово. Лишние пробелы при необходимости удалите с помощью команды ″**Г**.

5.3. В состоянии поиска данных введите команду ″**Ц**. В результате на экране появится текст последней карточки вашей картотеки. Печать карточки обеспечьте вводом команды ″**П**. Обратите внимание: в получаемую при этом распечатку войдет также служебная информация КАРТ1715.

5.4. Воспользуйтесь операцией поиска, задавая в качестве образца текста ключевое слово, входящее только в данную карточку. Учитывайте при этом, что в основном режиме поиска просматриваются только индексированные слова карточек.

5.5. Вы можете обеспечить при поиске данных просмотр и неиндексированных слов, для чего воспользуйтесь командой **МА** с последующим выполнением обычной операции поиска. Имейте в виду, что такой режим поиска данных потребует больше времени.

5.6. Предварительно создайте дополнительный файл формата карточки всего из одного поля. При работе с данными в картотеке введите команду **ФО**, а затем с помощью команды **Ф** задайте имя файла дополнительного формата. В дальнейшем вывод на экран и печать будет производиться в новом формате, в котором предусматривается вывод лишь одного поля карточек.

5.7. Находясь в SCP, удалите с диска файлы **библи.фмт** и **библи.фил**: а>ера библи.* **ЕТ**.



СИСТЕМА УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ РЕБУС



6.1. ДЛЯ ЧЕГО НУЖНА БАЗА ДАННЫХ?

Попробуем разобраться в том, когда вам может потребоваться создание базы данных и чем система управления базой данных отличается от других пакетов программ вроде электронной таблицы или электронной картотеки.

База данных

Когда говорят о базе данных, то под ней понимают совокупность взаимосвязанных массивов данных, хранящихся в памяти ЭВМ. Такое или подобное определение вы можете найти в работе [9] и других книгах, посвященных базам данных. Например, база данных, в которой хранится информация о какой-либо организации, может содержать перечень ее подразде-

лений, штатное расписание, список сотрудников, данные о работах, проводимых в организации, и т. д. Вся эта разнообразная информация составляет единую базу данных.

Система управления базой данных

Прежде чем вы сможете воспользоваться информацией, хранимой в базе данных, необходимо обеспечить ее ввод в память ЭВМ. Далее вам потребуется находить информацию в базе данных, обрабатывать ее тем или иным образом, выводить результаты обработки на дисплей, печатающее устройство или в базу данных, дополнять базу данных новой информацией. Всю эту работу выполняет система управления базой данных.

У вас может возникнуть вопрос: чем система управления базой данных отличается от любой другой программы, обрабатывающей дисковые файлы? Принципиальное различие состоит в том, что система управления базами данных не зависит от той информации, которая хранится в базе данных. Если же вы написали программу, обрабатывающую какой-то определенный файл, например, на языке Паскаль, то вам не удастся воспользоваться этой программой для обработки другого файла.

Еще одна возможность, предоставляемая системой управления базами данных, связана с тем, что базой данных можете пользоваться не только вы, но и ваши коллеги. Естественно, что у каждого пользователя могут существовать собственные потребности в информации. Система управления базой данных позволит организовать для каждого пользователя доступ к интересующим его данным и избавит его от необходимости знать, какая еще информация хранится в базе данных.

Воспользоваться этими возможностями системы управления базой данных можно с помощью ее команд, последовательности которых вы можете записать в командный файл на диск. Это позволит вам многократно автомати-

чески выполнять нужные действия по обработке базы данных. Освоение команд системы управления базой данных равноценно освоению языка программирования. В работе [9] об этом сказано следующее: «Потребителей микро-ЭВМ пытаются убедить, что это некоторым волшебным образом не языки программирования, и поэтому пакеты программ управления базами данных рассчитаны на наивного и неопытного „неподготовленного" пользователя. На самом деле это такие же языки программирования, как и любые другие, и вы не решите с их помощью свои задачи, если не обладаете опытом и настойчивостью программиста.». Тем не менее это не должно вас пугать. Освоение любого пакета программ требует бóльших или меньших затрат времени. Все зависит от вашего упорства и оптимизма.

Что лучше — РЕБУС, ВАРИТАБ или КАРТ1715 на Роботроне 1715?

Теперь попытаемся сравнить систему управления реляционной базой данных с электронной таблицей и электронной картотекой. Начнем с электронной картотеки. Согласно работе [22] «диагноз» может быть следующим: ни на что, кроме картотеки, созданной по образу и подобию библиотечной, она не годится. Что касается электронной таблицы, то здесь дело обстоит несколько сложнее. Что в принципе невозможно сделать, используя этот пакет программ? Во-первых, электронная таблица не позволяет осуществлять выборку информации из таблицы, а во-вторых, невозможно произвести сортировку строк таблицы в соответствии со значениями некоторого столбца (например, упорядочить по алфавиту фамилии сотрудников в случае расчета ведомости начисления заработной платы). С помощью пакета программ электронной таблицы вы можете обработать совместно столько таблиц, сколько сможете разместить в рабочем поле. Однако такие манипуляции напоминают попытку проникнуть в дом через

дымоход, когда никто не мешает войти в открытую дверь — воспользоваться системой управления базой данных.

К какому выводу мы с вами пришли?

Все эти три пакета программ решают одну и ту же задачу — в том или ином виде они формируют и обрабатывают таблицы. Однако делают они это с разным успехом. В этом плане наибольшими возможностями обладает РЕБУС. Это значит, что, пользуясь им, вы можете решить те задачи, которые позволяют решать и ВАРИТАБ, и КАРТ1715, а также более сложные. Расплата за универсальность РЕБУСа — большие затраты времени на его освоение. В конечном счете выбор пакета программ определяется тем, что именно вам требуется сделать. Очень хорошо, если для решения задачи вам будет достаточно ВАРИТАБа или КАРТ1715. Эти пакеты более удобны для пользователей, профессиональные интересы которых лежат в стороне от проблем программирования. Другое дело РЕБУС. Освоение и решение задач с применением этого пакета программ стоит куда бóльших усилий. Поэтому РЕБУС — средство, скорее, для профессиональных программистов, чем для неподготовленных пользователей.

6.2. ЧТО ЭТО ТАКОЕ — РЕБУС?

Происхождение РЕБУСа

Если вас интересует возможность выбора системы управления базой данных, то на Роботроне 1715 ничего другого нет. Кроме РЕБУСа либо того, что может называться по-другому, но подобно РЕБУСу является ничем иным, как системой управления базой данных dBASE II, вы не найдете ничего ни лучше, ни хуже. Тем самым наши организации — поставщики программного обеспечения опровергают старую истину — насильно мил не будешь.

Что касается метода получения РЕБУСа из dBASE II, то он достаточно прост. В американском оригинале переводятся с английского языка на русский имеющиеся сообщения. В результате вы можете стать счастливым обладателем отечественного пакета программ вместе с документацией к нему [17].

Состав системы управления базой данных РЕБУСа

РЕБУС поставляется на двух гибких дисках Таллинским научно-учебным центром. На первом диске записаны файлы, составляющие РЕБУС, на втором — файлы, предназначенные для выполнения контрольных примеров. Для работы с системой управления базой данных необходимы два файла с первого диска: REBUS.COM (занимает 30 Кбайт дисковой памяти); REBUSOVR.COM (40 Кбайт). Все остальные файлы не являются жизненно необходимыми для работы системы управления базой данных.

Файл REBUSMSG.TXT (44 Кбайт) позволит вам получить помощь в использовании команд и функций системы управления базой данных непосредственно во время работы с ней.

Файлы MAKET.COM (14 Кбайт), ZSCRN.OVL (20 Кбайт), DGEN.OVL (8 Кбайт) являются составными частями программы создания форм для организации ввода/вывода информации.

Командный файл RSORT.CMD (4 Кбайт) предназначен для сортировки записей в порядке русского алфавита.

Командный файл SETS.CMD (2 Кбайт) служит для установки начальных значений параметров системы управления базой данных.

Перечисленные выше файлы занимают на диске 152 Кбайт. Это значит, что вы сможете разместить их все на одном диске вне зависимости от варианта исполнения Роботрона 1715.

6.3. ПОСТАВИМ ЗАДАЧУ

Определите нужную вам информацию

Никакой компьютер не сможет помочь вам, если вы не знаете, какая помощь требуется и в чем она должна заключаться. Представьте, что вам пришлось стать главой некой организации, занимающейся разработкой программного обеспечения, причем все сотрудники не знают ничего кроме Бейсика. Ситуация — достаточно традиционная. Как правило, она усугубляется тем, что «знатоки» Бейсика не хотят слышать ни о чем другом.

Для начала вы решаете разобраться в том, что может дать вашей организации использование баз данных. Но заниматься таким делом без конкретной задачи бесполезно. Самое простое — попробовать самому создать базу данных, в которой бы хранилась информация о самой организации, необходимая вам лично в работе.

Первое, что вы выясняете, — какие подразделения есть в организации и кто ими руководит. Немаловажно также, что делают эти подразделения, когда наступают сроки окончания работ, кто является заказчиком, сколько можно выплатить денег исполнителям за год и кто руководит работами. Что касается самих сотрудников, то надо знать фамилию, имя и отчество каждого из них, в каком из подразделений работает, какую должность занимает, какой оклад получает и в какой работе участвует.

Таким образом вы определили состав хранимых данных, которые впоследствии при необходимости всегда можно будет дополнить.

Как выглядит база данных на бумаге?

Если вы попытаетесь представить всю требующуюся информацию в одной таблице, то ее размеры, вероятнее всего, станут для вас не очень приятным сюрпризом. Кроме того, от вас потребуется огромное терпение, когда нужно будет заполнить эту таблицу. Ведь для каждого

ПОДРАЗДЕЛЕНИЯ

Код подразделения	Наименование подразделения	Код сотрудника
1	Отдел бессистемного программирования	3
2	Отдел АСУ чем попало	4

ДОЛЖНОСТИ

Код должности	Наименование должности
1	Ведущий инженер
2	Старший инженер
3	Инженер-программист

СОТРУДНИКИ

Код сотрудника	Фамилия, имя, отчество	Код		Оклад	Код работы
		подразделения	должности		
1	Иванов Иван Иванович	2	1	250	2
2	Дубова Татьяна Петровна	2	3	130	2
3	Кобрин Сергей Тимофеевич	1	2	150	1
4	Петров Петр Петрович	2	1	250	3
5	Дурова Юлия Сергеевна	1	3	130	1
6	Сидоров Сидор Сидорович	2	1	250	3

РАБОТЫ

Код работы	Наименование работы	Срок окончания	Заказчик	Фонд заработной платы	Код сотрудника
1	Разработка пакета программ «Шлеп-Хлоп»	31/12/88 г.	Завод им. 1 апреля	4800	3
2	Разработка задумчивого интерфейса к базе данных	30/06/89 г.	ПО «Февраль»	5400	4
3	Разработка АСУ производством	31/12/88 г.	Трест Главщепка	6000	1

Рис. 6.1. Пример базы данных, представленной несколькими таблицами

сотрудника организации надо указать полностью наименование отдела, в котором он работает, наименование работы, в которой этот сотрудник принимает участие, а также всю информацию, относящуюся к этой работе.

Вывод, к которому вы вскоре придете, заключается в том, что одна и та же информация будет многократно повторяться в таблице. Избежать этого вы можете, воспользовавшись несколькими таблицами, выделяя повторяющиеся элементы в отдельные таблицы. В результате вы получите нечто подобное тому, что представлено на рис. 6.1.

Рассмотрим некоторые связи, существующие между таблицами в рассматриваемом примере. В таблице ПОДРАЗДЕЛЕНИЯ в столбце «Код сотрудника» находятся значения, определяющие, кто из сотрудников организации руководит соответствующим отделом. Предположим, нас интересует, кто руководит отделом с кодом отдела 2. Выясняется, что это некто, значащийся под кодом сотрудника 4. Выяснить, кто это, можно, обратившись к таблице СОТРУДНИКИ. В ней надо найти строку, в которой в столбце «Код сотрудника» записано 4. Тут мы обнаруживаем, что это Петров Петр Петрович.

На рис. 6.1 все связи такого рода показаны стрелками. Что это нам дает? А то, что совершенно не обязательно одну и ту же информацию полностью дублировать везде, где она требуется. Достаточно поместить там тот элемент, который однозначно определяет всю совокупность данных. В рассматривавшемся примере значение в столбце «Код сотрудника» однозначно соответствует некоторому конкретному человеку. Поэтому можно использовать код сотрудника везде, где требуется указать, что речь идет о каком-то конкретном человеке. Аналогично тому, как это делалось в таблице ПОДРАЗДЕЛЕНИЯ, в таблице РАБОТЫ код сотрудника определяет руководителя работ.

Приведенный выше анализ сильно упрощен. Ничего не говорится о том, какая информация, хранящаяся в базе

данных, будет нужна для решения каждой конкретной задачи. И вообще ничего не говорится о задачах, которые будут использовать базу данных. Тем не менее подобный анализ потребуется от вас до того, как будет запущена система управления базой данных и вы приступите к созданию базы данных. В реальной ситуации на решение этих вопросов может быть затрачено несколько месяцев работы.

6.4. НЕМНОГО ИЗ ТЕОРИИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

6.4.1. ЧТО НАЗЫВАЮТ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ?

Таблица

Человечество начало использовать таблицу для представления данных задолго до появления реляционных баз данных и сопутствующей им теории. Каждому из нас в своей жизни приходилось составлять не одну таблицу, и едва ли кто-то отважится выразить свое непонимание по поводу того, что же такое таблица и как ею пользоваться. Использование таблиц позволяет представлять любые данные в обозримом виде и понятной форме. Таблица обладает следующими свойствами [11]:

- каждый элемент таблицы представляет собой один элемент данных, повторяющиеся группы отсутствуют;
- все столбцы в таблице однородные; это означает, что элементы каждого столбца имеют одинаковую природу;
- столбцам однозначно присвоены имена;
- в таблице нет двух одинаковых строк;
- в операциях с таблицей ее строки и столбцы могут просматриваться в любом порядке и любой последова-

тельности безотносительно к их информационному содержанию и смыслу.

Реляционная база данных

При работе с реляционной базой данных вам наверняка придется столкнуться с различной терминологией. Соответствие различных терминов вы можете найти в табл. 6.1. В дальнейшем для облегчения понимания будем пользоваться традиционной терминологией и той, которая принята в РЕБУСе.

Таблица 6.1

Соответствие терминов в теории и практике реляционных баз данных

Традиционные понятия	Теория реляционных баз данных	Система управления базой данных РЕБУС
Таблица	Отношение	Файл базы данных
Строка	Кортеж	Запись
Столбец	Атрибут	Поле
Множество допустимых значений элементов столбца	Домен	Тип и длина поля

По своей сути реляционная база данных представляет собой несколько связанных друг с другом таблиц. Однако никто, конечно, не заставляет вас делать несколько таблиц там, где может быть достаточно одной. Ваша задача заключается в определении состава столбцов каждой используемой таблицы и организации взаимосвязей таблиц, что обсуждалось уже в параграфе 6.3 при постановке задачи.

Работа с системой управления реляционной базой данных требует знания средств работы с таблицами. К этим средствам относятся восемь операций, которые должна

выполнять система управления реляционной базой данных для того, чтобы таковой называться по праву. Предлагаем вам ознакомиться с тем, что они собой представляют.

6.4.2. ОПЕРАЦИЯ ПРОЕКЦИИ

Операция проекции позволит вам выделить из таблицы один или более столбцов и получить вторую таблицу, содержащую только нужные вам столбцы. Обратившись к рис. 6.2, вы увидите, что получится, если из таблицы СОТРУДНИКИ (см. рис. 6.1) с помощью операции проекции выделить столбцы «Код подразделения» и «Код должности». В исходной таблице СОТРУДНИКИ содержатся шесть строк, из них строки 2, 3, 5 уникальны, а строки 4, 6 повторяют содержание первой строки. Операция проекции в ходе создания новой таблицы включает в свой состав только уникальные строки (2, 3, 5) исходной таблицы и один экземпляр повторяющихся строк (1, 4, 6).

СОТРУДНИКИ

Код подразделения	Код должности
2	1
2	3
1	2
2	1
1	3
2	1

РЕЗУЛЬТАТ ПРИМЕНЕНИЯ ОПЕРАЦИИ ПРОЕКЦИИ

Код подразделения	Код должности
2	1
2	3
1	2
1	3

Рис. 6.2. Выполнение операции проекции

Смысл примера заключается в том, что в результате будет получена таблица, в которой будут перечислены коды подразделений и коды должностей, которые есть в этих подразделениях.

Переходя от частного к общему, можно сформулировать следующее правило выполнения операции: результирующая таблица отличается от исходной тем, что она содержит только те столбцы, для которых выполняется операция проекции. Кроме того, в результирующей таблице должны отсутствовать повторяющиеся строки.

6.4.3. ОПЕРАЦИЯ ОГРАНИЧЕНИЯ

Эта операция дает вам возможность выделить из таблицы те строки, которые удовлетворяют некоторому поставленному вами условию. При этом исходная таблица и таблица, являющаяся результатом, имеют одинаковый состав столбцов.

Применим операцию ограничения к таблице СОТРУДНИКИ (см. рис. 6.1). Выделим из нее строку, применительно к которой значение в столбце «Оклад» равно 150 (табл. 6.2).

Таким образом, в результате выполнения операции ограничения вы получите новую таблицу, содержащую те же столбцы, что и исходная. Из исходной таблицы в резуль-

Таблица 6.2

Результат выполнения операции ограничения

Код сотрудника	Фамилия, имя, отчество	Код подразделения	Код должности	Оклад	Код работы
3	Кобрин Сергей Тимофеевич	1	2	150	1

тирующую переписываются только те строки, значения столбцов в которых удовлетворяют условиям, указанным в операции ограничения.

6.4.4. ОПЕРАЦИЯ СОЕДИНЕНИЯ

Эта операция позволит вам соединить две таблицы и получить третью. Легче всего показать действие этой операции на примере, приведенном в табл. 6.3 применительно к таблицам ПОЗДРАВЛЕНИЯ и СОТРУДНИКИ (см. рис. 6.1).

Таблица 6.3

Пример выполнения операции соединения

Код подразделения	Наименование подразделения	Код сотрудника	Фамилия, имя, отчество	Код должности	Оклад	Код работы
1	Отдел бессистемного программирования	3	Кобрин Сергей Тимофеевич	2	150	1
2	Отдел АСУ чем попало	4	Петров Петр Петрович	1	250	3

Таблицы, которые используются при выполнении операции соединения, содержат два общих столбца: «Код подразделения» и «Код сотрудника». Результирующая таблица содержит те столбцы, которые есть хотя бы в одной из таблиц. В нее будут записаны строки, полученные присоединением строки из таблицы ПОДРАЗДЕЛЕНИЯ к строке из таблицы СОТРУДНИКИ. Дополнительное

свойство, которым должны обладать соединяемые в рассматриваемом примере строки, таково: значения, находящиеся в столбцах «Код подразделения» этих двух таблиц, должны быть равны. То же самое условие должно выполняться в отношении столбцов «Код сотрудника» таблиц ПОДРАЗДЕЛЕНИЯ и СОТРУДНИКИ.

С учетом этих условий могут быть соединены строка 1 таблицы ПОДРАЗДЕЛЕНИЯ со строкой 3 таблицы СОТРУДНИКИ и строка 2 таблицы ПОДРАЗДЕЛЕНИЯ со строкой 4 таблицы СОТРУДНИКИ.

Таким образом получается, что операция соединения использует только те строки исходных таблиц, в общих столбцах которых содержатся равные значения.

Если эту операцию применить к таблицам, не имеющим общих столбцов, то она будет эквивалентна операции прямого произведения, с которой вы ознакомитесь несколько позже.

6.4.5. ПРОЧИЕ ОПЕРАЦИИ

Отвлечемся теперь от нашего примера и рассмотрим остальные операции на абстрактных примерах. В них для обозначения столбцов будем использовать прописные латинские буквы, а для обозначения значений в столбцах — строчные буквы с числовыми индексами.

Примеры этих операций приведены на рис. 6.3. Они не более чем иллюстрируют действия операции на некотором конкретном примере. Если вас заинтересуют детали, то вам следует обратиться к литературным источникам [4, 12, 13], что позволит ознакомиться с математическим аппаратом, используемым в теории реляционных баз данных [13], с системой управления реляционной базой данных [4]. Если вам захочется получить достаточно полное

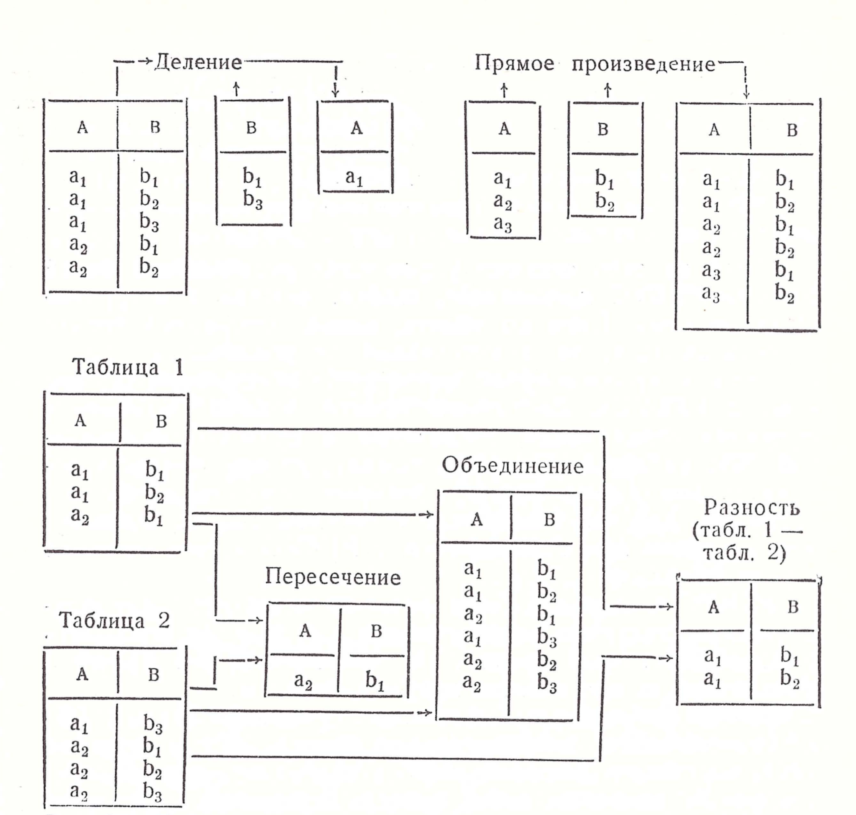


Рис. 6.3. Примеры выполнения операций деления, прямого произведения, пересечения, объединения и разности

представление о состоянии теории реляционных баз данных, то вам не обойтись без работы [12]. В каждой из этих книг вы сможете найти формальное описание операций, рассмотренных выше.

6.4.6. ЯВЛЯЕТСЯ ЛИ РЕБУС СИСТЕМОЙ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ?

Сама по себе система управления реляционной базой данных РЕБУС напоминает, скорее, детский конструктор «Сделай сам». Если вы знаете, какой операцией хотите воспользоваться, то практическая ее реализация будет состоять в использовании некоторой последовательности команд РЕБУСа. Достаточно просто можно выполнить операции ограничения, соединения и прямого произведения. Что касается остальных команд, то вам придется создавать командные файлы, которые выполняли бы эти операции. Так что знакомство с теорией, скорее, приведет вас в уныние при столкновении с РЕБУСом, чем поможет в решении задач.

РЕБУС и ему подобные системы называют полуреляционными, частично реляционными и т. п. Если подходить с подобной точкой зрения, то автомобиль без рулевого управления можно назвать полуавтомобилем.

Другое неудобство, связанное с использованием РЕБУСа, состоит в большом числе команд (62) и функций (14), не считая операций, что отнюдь не способствует ускорению освоения этого пакета программ. Ситуация становится для пользователя сравнима по удобствам с той, когда в упомянутом выше полуавтомобиле установлены микропроцессорная система управления движением, система речевого управления, система кондиционирования воздуха и т. д. Однако что толку от всего этого, если ездить на нем можно только прямо.

В работе [22] сказано просто, что, называя dBASE II (а значит, и РЕБУС) системой управления реляционной базой данных, желаемое выдают за действительное. Дру-

гой интересный факт заключается в том, что фирмы — разработчики программного обеспечения отказались от разработки систем управления реляционной базой данных для 8-разрядных микроЭВМ, считая это технически невозможным [22].

Тем не менее, какой бы несовершенной ни была эта система управления реляционной базой данных, опыт показал ее пригодность для решения большого числа практических задач.

6.5. РАБОТА С СИСТЕМОЙ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ РЕБУСа

6.5.1. ЧТО НАДО СДЕЛАТЬ ПЕРЕД ЗАПУСКОМ РЕБУСа?

Типы данных, используемые в РЕБУСе

К этому моменту нам известно, какую информацию и каким образом требуется хранить в базе данных. Но этого недостаточно для того, чтобы начать работу с системой управления базой данных. В чем тут дело?

Первое, о чем вам надо помнить, — это то, что система управления базой данных обеспечивает возможность использования ограниченного числа типов данных: символьного, числового и логического.

Данные символьного типа (обозначение в РЕБУСе — С) позволяют формировать, сохранять и обрабатывать последовательности символов — букв, цифр, знаков препинания, специальных символов.

Числовые данные (N) — это целые или вещественные значения, представленные в десятичной системе счисления. Они могут включать знак числа. Вещественные числа могут содержать точку, отделяющую целую часть числа от дробной. Вы должны запомнить, что строка «123» и число 123 — разные вещи.

Данные логического типа (L) позволяют представлять только два вида значений — ИСТИНА и ЛОЖЬ (ДА/НЕТ).

Вернемся к табл. 6.1, к той ее части, которая касается РЕБУСа. Приведенные в ней элементы — файл базы данных и поле — должны иметь имена. На вас ложится обязанность дать имена всем файлам базы данных и полям, которые будут использоваться.

Имя файла базы данных

Имя файла базы данных может содержать от одного до восьми символов. Допустимыми являются латинские буквы и цифры. Можно использовать и другие символы в имени базы данных. При этом действуют ограничения, налагаемые операционной системой на имя файла.

Обратимся снова к примеру и дадим имена файлам базы данных, в которых будет храниться информация, представленная ранее в таблицах. Сделаем это в виде табл. 6.4.

Имя поля

Файл базы данных содержит одну или несколько записей или не содержит их вовсе. Каждая запись состоит из одного или более полей. Каждому полю должно быть присвоено имя, включающее от одного до десяти символов — латинских букв, цифр, двоеточий. Имя поля начинается с буквы и не должно заканчиваться двоеточием.

Когда вами определены имена полей для записей каждого файла базы данных, остается сделать последний подготовительный шаг. Вы должны установить типы данных для представления информации в полях записи и максимальное количество символов (для символьных данных) или цифр (для числовых), которые могут быть использованы для этого.

Доведем до конца подготовку к вводу описания структуры базы данных. Все, что нам будет нужно, представим в виде табл. 6.5.

Теперь мы окончательно готовы к работе с РЕБУСом.

Таблица 6.4

Соответствие имен таблиц и имен файлов базы данных

Имя таблицы	Имя файла базы данных
ПОДРАЗДЕЛЕНИЯ	DIV
РАБОТЫ	WRK
СОТРУДНИКИ	EMP
ДОЛЖНОСТИ	OFF

Таблица 6.5

Описание структуры базы данных

Имя файла базы данных	Имя поля	Тип	Длина
DIV	DIV:CODE	C	2
	DIV:NAME	C	40
	EMP:CODE	C	3
WRK	WRK:CODE	C	2
	WRK:NAME	C	40
	WRK:END	C	8
	WRK:CUST	C	20
	WRK:FUND	N	5
	EMP:CODE	C	3
EMP	EMP:CODE	C	3
	EMP:NAME	C	30
	DIV:CODE	C	2
	OFF:CODE	C	2
	EMP:TAX	N	3
	WRK:CODE	C	2
OFF	OFF:CODE	C	2
	OFF:NAME	C	20

6.5.2. ЗАПУСК РЕБУСа

Произвести запуск РЕБУСа вы сможете только в том случае, когда загружена операционная система. Диск с РЕБУСом должен быть установлен в дисковод, который является текущим активным в данный момент времени. В противном случае вы должны сделать текущим активным тот дисковод, в котором установлен диск с РЕБУСом.

Если вы заменяли диски, то не забудьте произвести сброс дисковой системы. Для этого нужно одновременно нажать клавиши **CTRL** и **C**. Забывчивость в подобной ситуации может привести к потере результатов вашей работы.

Теперь вы можете запустить РЕБУС, введя с клавиатуры REBUS и нажав затем клавишу **ET**.

Далее РЕБУС попросит вас ввести текущую дату. Это будет сделано в виде

ВВЕДИТЕ ДАТУ ИЛИ <ET>
(ДД/ММ/ГГ):

Ввести дату вы можете несколькими способами. Например, ввод в качестве даты 1 апреля 1988 г. может быть выполнен следующими способами:

1/04/88
1 4 88
01.04.88
01:4:88

Ввод даты завершается нажатием клавиши **ET**. Если вы просто нажмете клавишу **ET**, не вводя дату, то будет автоматически установлена дата 00/00/00.

После ввода даты будет выведено сообщение

ВВЕДИТЕ 'HELP', 'HELP REBUS' ИЛИ КОМАНДУ

Появление точки в начале строки означает, что система управления базой данных готова к выполнению команд.

6.5.3. ВВОД ОПИСАНИЯ СТРУКТУРЫ ФАЙЛА БАЗЫ ДАННЫХ И ЕГО ЗАПОЛНЕНИЕ

Ввод описания структуры файла базы данных

Ввести описание структуры файла базы данных вы можете, воспользовавшись командой CREATE. Ниже приведен пример ввода описания структуры файла базы данных EMP, определенной нами в табл. 6.5.

```
. CREATE
ВВЕДИТЕ ИМЯ ФАЙЛА: EMP
ВВЕДИТЕ ФОРМАТ СОГЛАСНО ПРИМЕРУ:
 ПОЛЕ      ИМЯ,ТИП,ДЛИНА,ДЛИНА ДРОБНОЙ ЧАСТИ
 001      EMP:CODE,C,3
 002      EMP:NAME,C,30
 003      DIV:CODE,C,2
 004      OFF:CODE,C,3
 005      EMP:TAX,N,3
 006      WRK:CODE,C,2
 007
ВВОД ДАННЫХ СЕЙЧАС? Y

ЗАПИСЬ 00001

EMP:CODE  : 1
EMP:NAME  : ИВАНОВ ИВАН ИВАНОВИЧ
DIV:CODE  : 2
OFF:CODE  : 1
EMP:TAX   : 250
WRK:CODE  : 2

ЗАПИСЬ 00002

EMP:CODE  :
```

Если, введя команду CREATE, вы не указали имя файла базы данных, то РЕБУС запросит его ввод, как это

показано выше. Если файл базы данных с указанным вами в команде CREATE именем уже существует, система управления базой данных предоставит вам возможность выбора одного из двух: уничтожить существующий файл базы данных и создать новый с указанным именем либо ввести другое имя для создаваемого файла базы данных.

Когда вами указано имя создаваемого файла базы данных, РЕБУС начнет ввод описания его структуры. Сначала будет выведен номер очередного поля и курсор останется в той же строке. Теперь вы должны ввести описание поля в следующей последовательности: имя поля, его тип, максимальная длина поля. Для числового поля может быть указана длина дробной части. Эти элементы описания должны быть разделены запятыми. Ввод описания очередного поля завершается нажатием клавиши ET. Затем вы можете ввести описание очередного поля или завершить ввод описания структуры файла базы данных. Для завершения ввода описания структуры нажмите клавишу ET в ответ на запрос ввода описания очередного поля. Это и было сделано в примере в ответ на запрос ввода описания седьмого поля.

Заполнение файла базы данных

Возможности команды CREATE не исчерпываются вводом описания структуры файла базы данных. Завершив его, вы можете внести информацию в созданный файл базы данных. Впрочем, никто не торопит вас с этим. Система управления базой данных задаст вопрос: желаете ли вы сейчас ввести данные или нет? Если вы нажмете клавишу N, то это будет означать отказ от такой возможности. Нажатие клавиши Y позволит вам ввести данные, что и продемонстрировано в приведенном примере.

Система управления базой данных выводит номер записи файла базы данных, в которую будет производиться ввод (ЗАПИСЬ 00001). Затем будет выведено имя пер-

вого поля, за которым ставится двоеточие. Курсор останется в этой же строке. Теперь РЕБУС готов получить то, что вы намереваетесь записать в это поле. Ввод данных в поле завершится по нажатию клавиши **ЕТ**. Затем РЕБУС выведет имя следующего поля и будет ожидать ввода данных в него.

И так поле за полем, запись за записью вы будете заполнять файл базы данных. Как вы могли заметить, порядок запроса ввода данных в поля совпадает с порядком их следования при вводе описания структуры файла базы данных. Завершить процесс заполнения файла базы данных вы можете только в момент, когда запрашивается ввод в первое поле очередной записи. Для этого вам достаточно нажать клавишу **ЕТ**, не вводя больше ничего с клавиатуры.

Проверьте состояние файла базы данных

Всегда полезно убедиться в том, что вы потратили время не напрасно и не наделали ошибок. Простейший способ убедиться в этом показан ниже,

```
. USE EMP
. DISPLAY STRUCTURE
СТРУКТУРА ФАЙЛА:   A:EMP      .DBF
ЧИСЛО ЗАПИСЕЙ:     00001
ДАТА ПОСЛЕДНЕГО ОБНОВЛЕНИЯ: 29/07/88
ПЕРВИЧНАЯ БАЗА ДАННЫХ
ПОЛЕ       ИМЯ          ТИП  ДЛИНА   ДРОБНАЯ ЧАСТЬ
001      EMP:CODE        C     003
002      EMP:NAME        C     030
003      DIV:CODE        C     002
004      OFF:CODE        C     003
005      EMP:TAX         N     003
006      WRK:CODE        C     002
** ВСЕГО **                  00044
. DISPLAY ALL
00001  1   ИВАНОВ ИВАН ИВАНОВИЧ                2  1     250 2
```

Системе управления базой данных неизвестно, с каким файлом базы данных вы намереваетесь работать. «Объяснить» ей это можно с помощью команды USE, за которой через пробел следует имя нужного вам файла базы данных. Ввод команды завершается нажатием клавиши ET. Теперь ничто не мешает вам полюбоваться на дело своих рук:

◆ по команде DISPLAY STRUCTURE на экран дисплея выводятся имя файла базы данных, с которым вы сейчас работаете, количество записей в нем, дата его последнего обновления и структура записи в нем;

◆ по команде DISPLAY ALL на экран дисплея будут выведены все записи, внесенные к этому времени в файл базы данных.

Дополнение файла базы данных

Сколько бы вы ни внесли записей в файл базы данных, рано или поздно понадобится добавить в него еще что-то. Когда у вас возникает такая потребность, воспользуйтесь командой APPEND.

```
. APPEND

ЗАПИСЬ 00002

EMP:CODE   : 2
EMP:NAME   : ДУБОВА ТАТЬЯНА ПЕТРОВНА
DIV:CODE   : 2
OFF:CODE   : 3
EMP:TAX    : 130
WRK:CODE   : 2

ЗАПИСЬ 00003

EMP:CODE   :
```

Если вы сравните примеры использования команд CREATE и APPEND, то обнаружите, что «поведение»

команды APPEND аналогично тому, как «ведет себя» команда CREATE, когда вы используете ее для заполнения файла базы данных. Поэтому не будем останавливаться на этом более подробно.

6.5.4. ПОЧЕМУ ВАМ ПРИДЕТСЯ БОРОТЬСЯ С ОШИБКАМИ?

Неизбежность ошибок

Истории о том, к чему приводят ошибки в программах, достаточно часто можно встретить в специальной литературе по программированию. С появлением персональных ЭВМ положение усугубилось различными проявлениями компьютерного «хулиганства», от которого «достается» в основном базам данных. Конечно, трудно представить себе, что у нас кто-то сможет с помощью своей персональной ЭВМ открыть счет в Госбанке на миллиард, другой. Отставание в уровне развития средств вычислительной техники спасает нас от неприятностей такого сорта.

Однако не ошибается только тот, кто ничего не делает. При вводе данных в ЭВМ неизбежно будут возникать ошибки. А раз так, следует позаботиться, чтобы они не приводили к катастрофическим последствиям. Лучше всего обнаруживать и устранять ошибки сразу, в том месте, где они появились.

Целостность базы данных

Попробуем на примере выяснить, к чему может привести ошибка при вводе информации в базу данных. Обратимся для этого к рис. 6.1. Представим себе, что в таблице ПОДРАЗДЕЛЕНИЯ в первой строке записан код сотрудника не 3, а 7. Как вы можете заметить, сотрудни-

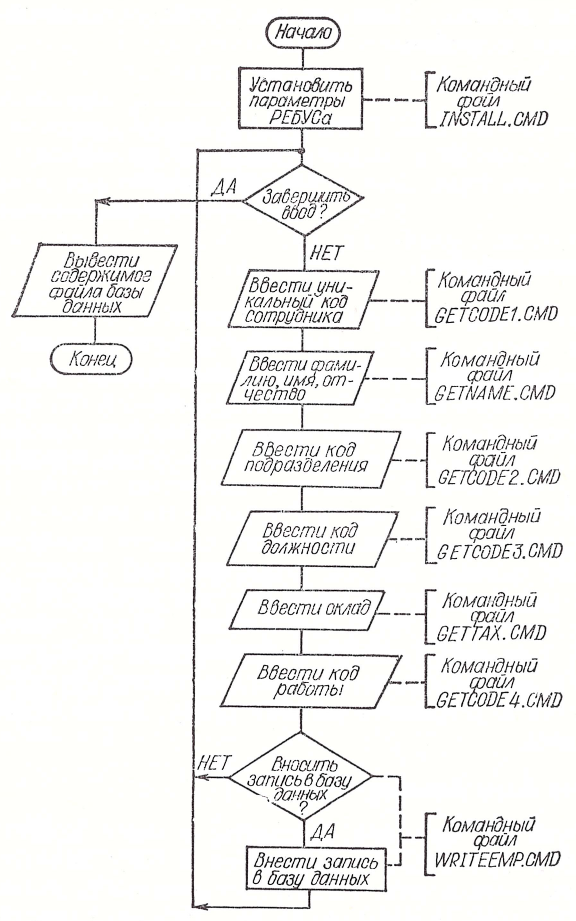


Рис. 6.4. Схема алгоритма внесения записи о сотруднике

ка с таким кодом нет. В результате такой ошибки вы никогда не сможете узнать, используя базу данных, кто в вашей организации руководит отделом бессистемного программирования. И это в лучшем случае. В худшем — руководителем этого отдела может «стать» любой сотрудник вашей организации. Все будет зависеть от того, каким образом обрабатывается информация из базы данных.

Ситуация, о которой шла речь, означает, что связи между данными в базе нарушены. А коли так, то, следуя определению, такую совокупность данных нельзя назвать базой данных. К аналогичным последствиям могут привести ошибки в любой другой таблице.

Существует, правда, вторая ситуация, когда вместо кода сотрудника 3 будет записано 2. Такую ошибку можно распознать и устранить, если вам известно, кто руководит отделом бессистемного программирования.

Но вернемся к первому случаю и разберемся, как с ним можно бороться. Рассмотрим наши возможные действия на примере таблицы СОТРУДНИКИ, которая имеет наибольшее число связей с другими таблицами. Для большей наглядности представим это в виде схемы алгоритма (рис. 6.4).

Что требуется сделать?

Как вы можете заметить, приведенный выше алгоритм слишком сложен, чтобы его можно было реализовать одной командой РЕБУСа. Однако есть возможность организовать выполнение нужных действий, воспользовавшись командными файлами. Напомним, что РЕБУС позволяет автоматически выполнять последовательности команд, предварительно записанные в дисковый файл. Такой файл называется командным, и вы можете создать его с помощью любого текстового редактора. Кроме того, в РЕБУСе имеются средства для создания и модификации командных файлов. Важное свойство РЕБУСа состоит в том, что один командый файл может инициировать вы-

полнение другого командного файла. После завершения выполнения второго командного файла продолжится выполнение первого. В этом механизме вы можете найти много общего в принципами использования подпрограмм в языках программирования.

Теперь сформулируем требованиям к организации ввода записи о сотруднике:

◆ присваиваемый новому сотруднику код должен быть уникален.

Если введен код сотрудника, уже содержащийся в базе данных, то необходимо затребовать его повторный ввод;

◆ при вводе кода подразделения, кода должности и кода работы следует обеспечить контроль за тем, что подразделение, должность, работа с указанным кодом существуют.

Ввод какого-либо кода, не содержащегося в базе данных, должен приводить к требованию его повторного ввода.

Все остальные приемы будут пояснены по мере рассмотрения командых файлов, решающих эту задачу.

Командный файл APPEMP.CMD

Запустив командный файл APPEMP. CMD, вы сможете внести столько записей о сотрудниках, сколько потребуется. В процессе работы будут выполняться все те проверки, которые мы обсуждали выше.

Запустить командный файл можно двумя способами:

◆ если РЕБУС уже запущен, то вы должны ввести команду DO APPEMP;

◆ если вы собираетесь это сделать сразу после запуска РЕБУСа, то можно совместить запуск системы управления базой данных с выполнением нужного вам командного файла; для этого запуск РЕБУСа надо произвести командой REBUS APPEMP.

Если вы собираетесь проводить эксперименты с базой данных, о которой мы уже довольно долго говорим, вам

придется ввести описания структур всех ее файлов в соответствии с табл. 6.5 и заполнить файлы базы данных DIV, WRK и OFF той информацией, которая представлена на рис. 6.1.

Желание посмотреть в работе командные файлы потребует ввести их в том виде, в котором они приводятся в книге.

Посмотрим, что делает этот командный файл. Каждая его строка содержит одну команду системы управления базой данных. Если вам не удастся разместить команду

```
* ---------------------------------------
* КОМАНДНЫЙ ФАЙЛ APPEMP.CMD
* ---------------------------------------
DO INSTALL
STORE N TO CANCEL
* ---------------------------------------------------------------
* ЦИКЛ ВВОДА ДЛЯ ВНЕСЕНИЯ НОВЫХ ЗАПИСЕЙ О СОТРУДНИКАХ
* ЦИКЛ ЗАВЕРШАЕТСЯ ПО ВВОДУ ПУСТОЙ СТРОКИ В ПОЛЕ
* "КОД СОТРУДНИКА"
* ---------------------------------------------------------------
DO WHILE .NOT. CANCEL
   DO PUTFORM
   DO GETCODE1
   IF .NOT. CANCEL
      * -----------------------------
      * ВВОД НЕ БЫЛ ЗАВЕРШЕН
      * -----------------------------
      DO GETNAME
      DO GETCODE2
      DO GETCODE3
      DO GETTAX
      DO GETCODE4
      DO WRITEEMP
   ENDIF
ENDDO
ERASE
USE EMP
DISPLAY ALL
CLEAR
RETURN
```

в одной строке, то ее можно продолжить на следующей строке.

Итак, первая команда, с которой вам приходится столкнуться, — *. Она позволяет вносить комментарии в текст командного файла. Комментарии не оказывают никакого влияния на выполнение командного файла.

Реальная работа начинает выполняться с запуска командного файла INSTALL.CMD (см. стр. 258), устанавливающего параметры системы управления базой данных.

Логическая переменная памяти CANCEL требуется для организации цикла ввода записей о сотрудниках. Пока сохраняется ее значение .N. (NO), производится ввод записей. Присвоение этой переменной значения .Y. (YES) приведет к окончанию ввода.

Команда STORE N TO CANSEL устанавливает начальное значение .N. переменной памяти CANCEL. Это дает возможность начать цикл ввода и выполнить его по крайней мере один раз.

Для организации цикла используются две команды: DO WHILE .NOT. CANCEL и ENDDO. Команды, находящиеся между ними, будут выполняться, пока истинно выражение .NOT. CANCEL в команде DO WHILE. Это значит, что цикл выполняется, пока значение переменной памяти CANCEL равно .N.

Выполнение команд, входящих в рассматриваемый нами цикл, начинается с запуска командного файла PUTFORM.CMD (см. стр. 260). В результате его работы на экран дисплея будет выведена форма, которая подскажет вам, какую именно информацию требуется вводить. Наблюдаемая вами при этом картина будет похожа на ту, которая была при использовании команды APPEND.

Теперь все готово к вводу информации о сотруднике. Сначала вводится код сотрудника. Это делается с помощью командного файла GETCODE1.CMD (см. стр. 261). Если будет введен «пустой» код сотрудника, то перемен-

ная памяти CANCEL после выполнения этого командного файла будет содержать значение .Y. Это, как вы помните, означает прекращение ввода информации о сотрудниках.

Теперь надо проверить — прекратить или продолжать ввод. Ответ на этот вопрос содержится в переменной памяти CANCEL. Получить его можно, проверив значение CANCEL с помощью команды IF. Если выражение, указанное в этой команде (.NOT. CANCEL), истинно, то будет выполнена последовательность команд, находящаяся между командами IF и ENDIF. В противном случае оказывается, что в цикле больше нет команд, которые следовало бы выполнить, и проверка условия в команде DO приведет к его завершению, так как в переменной памяти CANCEL содержится значение .Y. и, следовательно, выражение в команде DO ложно.

Если не было ввода «пустого» кода сотрудника, то выполняется последовательность команд, заключенных между командами IF и ENDIF. Один за другим выполняются командные файлы:

- GETNAME.CMD (см. стр. 264) — ввод фамилии, имени и отчества сотрудника;
- GETCODE2.CMD (см. стр. 265) — ввод кода подразделения, в котором работает сотрудник;
- GETCODE3. CMD (см. стр. 266) — ввод кода должности сотрудника;
- GETTAX.CMD (см. стр. 268) — ввод оклада сотрудника;
- GETCODE4.CMD (см. стр. 267) — ввод кода работы, которую выполняет сотрудник;
- WRITEEMP.CMD (см. стр. 269) дает возможность отказаться от внесения записи данных либо внести ее в файл базы данных.

Вернемся к происходящему после окончания выполнения цикла внесения записей. Сначала очищается экран дисплея (команда ERASE). Затем на экран дисплея выводятся записи, содержащиеся в файле базы данных EMP.

Это делает уже известная вам последовательность команд USE EMP и DISPLAY ALL. Далее уничтожаются все переменные памяти, которые появились к этому моменту (одна из таких переменных — CANCEL), и закрываются все файлы базы данных, используемые в этот момент (команда CLEAR). Завершение выполнения командного файла происходит по команде RETURN. После этого РЕБУС перейдет в режим непосредственного ввода команд. Это значит, что далее вы будете вводить команды с клавиатуры, как это делалось, например, при вводе описания структуры файла базы данных EMP.

Теперь перейдем к рассмотрению командных файлов, упомянутых выше.

Командный файл INSTALL.CMD

Как вы уже знаете, командный файл INSTALL.CMD производит установку параметров системы управления базой данных. Эту работу выполняет команда SET.

```
* -----------------------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ INSTALL.CMD
* УСТАНОВКА ПАРАМЕТРОВ СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ
* -----------------------------------------------------------------
SET COLON OFF
SET TALK OFF
SET CONFIRM ON
RETURN
```

Что же делает этот командный файл и зачем это нужно?

Команда SET COLON OFF подавляет вывод двоеточий, ограничивающих те области экрана дисплея, в которых будет осуществляться ввод данных. Если эти двоеточия выводятся, то вся польза от них заключается в возможности оценить количество символов, которые вы еще можете ввести до заполнения поля. Большого смысла в этом

нет, поскольку система управления базой данных не позволит вам ввести символов больше, чем это определяется длиной поля.

Команда SET TALK OFF подавляет вывод на экран дисплея результатов выполнения команд. Естественно, это не означает, что команды ввода и отображения данных перестают работать. Просто на экран прекращается вывод результатов типа

4

после выполнения команды

STORE 2*2 TO NUM

Команда SET CONFIRM ON отменяет автоматическое окончание ввода информации при заполнении поля. Для окончания ввода в поле вы должны нажать клавишу **ET**. Пока она не будет нажата, имеется возможность изменить содержимое поля, находясь непосредственно в нем.

Команда RETURN обеспечивает возврат к выполнению командного файла, запустившего рассматриваемый нами командный файл.

Командный файл PUTFORM.CMD

Командный файл PUTFORM.CMD выводит на экран дисплея форму, подсказывающую вам, какую информацию требуется вводить. Если сравнить эту подсказку с той, которую дает при вводе команда APPEND, то придется признать, что, например, текст КОД СОТРУДНИКА вносит больше ясности, чем EMP:CODE.

Что же делает этот командный файл? Для начала очищается экран дисплея. Это делается командой ERASE. А дальше на экран дисплея выводятся шесть строк, поясняющих смысл той информации, которую вам следует ввести. Желаемый результат достигается использованием последовательности команд @. Как вы можете заметить, каждая из этих команд состоит из четырех частей:

◆ первое число указывает номер строки на экране дисплея, в которой будет выведен нужный вам текст.

Вы можете использовать 24 строки с номерами от 0 до 23. К строке с номером 0 лучше не прибегать, потому что РЕБУС может использовать ее для вывода сообщений в процессе ввода информации;

```
* -----------------------------------------------
* КОМАНДНЫЙ ФАЙЛ PUTFORM.CMD
* ВЫВОД ФОРМЫ ДЛЯ ВВОДА НА ЭКРАН ДИСПЛЕЯ
* -----------------------------------------------
ERASE
@ 4, 0 SAY '          КОД СОТРУДНИКА:
@ 6, 0 SAY 'ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО:
@ 8, 0 SAY '      КОД ПОДРАЗДЕЛЕНИЯ:
@ 10, 0 SAY '          КОД ДОЛЖНОСТИ:
@ 12, 0 SAY '                  ОКЛАД:
@ 14, 0 SAY '            КОД РАБОТЫ:
RETURN
```

◆ второе число, отделенное от первого запятой, определяет номер позиции в строке, начиная с которой будет выводиться текст.

Номер позиции в строке может изменяться от 0 до 79. В качестве начала отсчета принят левый верхний угол экрана дисплея. Его координаты — 0,0;

◆ SAY означает, что далее будет указан подлежащий выводу текст;

◆ все, что находится между апострофами, будет выведено на экран дисплея в указанной строке, начиная с указанной позиции в этой строке.

Командный файл GETCODE1.CMD

Рассмотрим теперь процесс ввода кода сотрудника и способы решения тех проблем, которые при этом возникают.

```
* ------------------------------------
* КОМАНДНЫЙ ФАЙЛ GETCODE1.CMD
* ------------------------------------
STORE Y TO EXIST
* --------------------------------------------------------
* ЦИКЛ ВВОДА КОДА СОТРУДНИКА
* ЦИКЛ ЗАВЕРШАЕТСЯ ПРИ ВВОДЕ КОДА СОТРУДНИКА,
* РАНЕЕ НЕ ИСПОЛЬЗОВАВШЕГОСЯ В БАЗЕ ДАННЫХ
* --------------------------------------------------------
DO WHILE EXIST
  STORE '   ' TO CODE:1
  @ 4, 24 GET CODE:1 PICTURE '999'
  READ
  IF CODE:1 <>
    USE EMP
    COUNT ALL FOR CODE:1 = EMP:CODE TO NUM
    IF NUM = 0
      * ----------------------------------------------------
      * ВВЕДЕН РАНЕЕ НЕ ИСПОЛЬЗОВАВШИЙСЯ КОД СОТРУДНИКА
      * ----------------------------------------------------
      STORE N TO EXIST
    ELSE
      @ 23, 0 SAY;
      'ОШИБКА: УКАЗАННЫЙ КОД СОТРУДНИКА УЖЕ ИСПОЛЬЗУЕТСЯ'
    ENDIF
  ELSE
    * ------------------------------------------------------
    * ВВЕДЕНА ПУСТАЯ СТРОКА - ПРИЗНАК ПРЕКРАЩЕНИЯ ВВОДА
    * ------------------------------------------------------
    STORE N TO EXIST
    STORE Y TO CANCEL
  ENDIF
ENDDO
@ 23, 0 SAY;.
'
RETURN
```

Напомним, что ввод кода сотрудника должен запрашиваться до тех пор, пока не будет введено значение, удовлетворяющее поставленному условию. Это значит, что процесс ввода носит циклический характер.

Для организации цикла воспользуемся переменной памяти логического типа EXIST. Если ее значение .Y., то

это значит, что в файле базы данных существует запись, удовлетворяющая некоторому условию, а при .N. такой записи нет. Сейчас нас будет интересовать запись, содержащая код сотрудника, совпадающий с введенным. Это значит, что цикл ввода кода сотрудника должен выполняться, пока значение переменной памяти EXIST равно .Y.

Что делается в цикле? Для начала устанавливается пустое значение переменной памяти символьного типа CODE:1, которая будет использоваться для хранения введенного кода сотрудника. Это делается командой STORE' 'TO CODE:1. Теперь можно ввести код сотрудника. Для этого служат команды @ и PEAD. Команда @ имеет вид, отличающийся от того, который был в командном файле PUTFORM.CMD. В данном случае координаты в команде @ указывают, в каком месте экрана будет производиться ввод. Остальные элементы команды имеют следующий смысл:

◆ GET указывает на то, что будет осуществляться ввод информации;
◆ CODE:1 определяет, что ввод будет производиться в переменную памяти с этим именем;
◆ PICTURE свидетельствует о том, что далее будет задан формат для ввода данных;
◆ '999' — собственно формат ввода данных — это то, что заключено в апострофы.

Он определяет, что должно быть введено не более трех цифр. На это указывают три символа 9. Если будут попытки ввода каких-либо символов, отличающихся от цифровых, то такие символы будут игнорироваться.

Команда READ выполняет ввод информации в соответствии с тем, как это было определено рассмотренной выше командой @.

После ввода информации в переменную памяти требуется определить, было ли введено что-либо или нет, что можно сделать с помощью команды IF. Если выражение CODE:1 <>' ' истинно, значит, был введен код сотруд-

ника. Теперь нужно определить, есть ли в файле базы данных EMP запись, поле EMP:CODE которой содержит значение, равное введенному коду сотрудника. Воспользуемся для этой цели командой COUNT, позволяющей подсчитать количество записей в базе данных, удовлетворяющих некоторому условию. Команда COUNT содержит следующие элементы:

◆ ALL задает область действия команды.

В данном случае анализируются все записи в файле базы данных;

◆ FOR означает, что далее будет указано условие, определяющее интересующие нас записи;

◆ CODE:1=EMP:CODE то условие, о котором идет речь; оно свидетельствует о том, что требуется определить количество записей, значение поля EMP:CODE которых совпадает с введенным кодом сотрудника;

◆ TO означает, что далее будет указана переменная памяти, в которую будет помещено определенное в результате выполнения команды количество записей;

◆ NUM — это имя переменной памяти числового типа, в которую будет помещен результат выполнения команды COUNT.

Если таких записей не найдено, о чем свидетельствует значение 0 переменной памяти NUM, значит, введен ранее не использовавшийся код сотрудника. Присвоение значения .N. переменной памяти EXIST в этом случае приведет к окончанию рассматриваемого нами цикла ввода. Значение переменной памяти NUM, отличное от 0, говорит о том, что введен недопустимый код сотрудника. В такой ситуации будет выведено сообщение об ошибке и запрошен повторный ввод кода сотрудника.

Имеет смысл обратить внимание на команду @ выводящую сообщение об ошибке. Эта команда размещается на двух строках. «Сказать» системе управления базой данных, что это одна команда, можно вводом точки с запятой в конце первой строки, в которой размещается команда.

А сейчас вернемся к тому, что требуется сделать, если введен «пустой» код сотрудника:

◆ во-первых, надо завершить цикл ввода кода сотрудника, поэтому переменной памяти EXIST присваивается значение .N.;

◆ во-вторых, необходимо завершить цикл ввода записей о сотрудниках.

Этот цикл подробно рассматривался, когда речь шла о командном файле APPEMP.CMD. Как вы помните, он завершается, если переменная памяти CANCEL содержит значение. Y. Именно поэтому переменной CANCEL здесь должно быть присвоено значение .Y.

По завершении цикла ввода кода сотрудника остается очистить ту область экрана дисплея, в которой могло находиться сообщение об ошибочном вводе кода сотрудника.

Командный файл GETNAME.CMD

Теперь нужно ввести фамилию, имя и отчество сотрудника.

```
* -------------------------------
* КОМАНДНЫЙ ФАЙЛ GETNAME.CMD
* -------------------------------
STORE '                              ' TO NAME
* ----------------------------------------------
* ВВЕСТИ ФАМИЛИЮ, ИМЯ, ОТЧЕСТВО СОТРУДНИКА
* ----------------------------------------------
@ 6, 24 GET NAME PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
RETURN
```

Вы, наверно, не находите ничего особенно нового в этом командном файле. Единственно, что может вызвать вопрос,— это формат, использованный в команде @ Тридцать символов X свидетельствуют о том, что в качестве фамилии, имени и отчества сотрудника вы можете ввести до 30 любых символов.

Командные файлы GETCODE2.CMD, GETCODE3.CMD, GETCODE4.CMD

Теперь необходимо решить задачу ввода кодов подразделения, должности и работы, относящихся к сотруднику, информация о котором вносится в базу данных.

```
* ---------------------------------
* КОМАНДНЫЙ ФАЙЛ GETCODE2.CMD
* ---------------------------------
STORE N TO EXIST
* ------------------------------------------
* ЦИКЛ ВВОДА КОДА ПОДРАЗДЕЛЕНИЯ
* ЦИКЛ ЗАВЕРШАЕТСЯ, ЕСЛИ ВВЕДЕН
* ИСПОЛЬЗУЮЩИИСЯ КОД ПОДРАЗДЕЛЕНИЯ
* ------------------------------------------
DO WHILE .NOT. EXIST
  STORE '  ' TO CODE:2
  @ 8, 24 GET CODE:2 PICTURE '99'
  READ
  USE DIV
  COUNT ALL FOR CODE:2 = DIV:CODE TO NUM
  IF NUM <> 0
    * ---------------------------------------
    * ВВЕДЕН ИСПОЛЬЗУЕМЫЙ КОД ПОДРАЗДЕЛЕНИЯ
    * ---------------------------------------
    STORE Y TO EXIST
  ELSE
    @ 23, 0 SAY;
    'ОШИБКА: УКАЗАННЫЙ КОД ПОДРАЗДЕЛЕНИЯ НЕ ИСПОЛЬЗУЕТСЯ
  ENDIF
ENDDO
@ 23, 0 SAY;

RETURN
```

Командный файл GETCODE2.CMD осуществляет ввод кода подразделения, где работает сотрудник, запись о котором вносится в настоящий момент времени.

Введенное значение сохраняется в переменной памяти CODE:2.

Здесь решается задача ввода такого кода подразделения, который уже содержится в файле базы данных DIV. Именно поэтому признаком окончания цикла ввода является наличие в этом файле базы данных записи, поле DIV:CODE которой содержит значение, совпадающее с введенным кодом подразделения. Кроме того, недопустим ввод «пустого» кода подразделения. В остальном работа командного файла GETCODE2.CMD аналогична работе GETCODE1.CMD.

Командный файл GETCODE3.CMD обеспечивает ввод кода должности сотрудника. Этот код должен содержаться в файле базы данных OFF.

```
* ------------------------------
* КОМАНДНЫЙ ФАЙЛ GETCODE3.CMD
* ------------------------------
STORE N TO EXIST
* --------------------------------
* ЦИКЛ ВВОДА КОДА ДОЛЖНОСТИ
* ЦИКЛ ЗАВЕРШАЕТСЯ, ЕСЛИ ВВЕДЕН
* ИСПОЛЬЗУЮЩИЙСЯ КОД ДОЛЖНОСТИ
* --------------------------------
DO WHILE .NOT. EXIST
  STORE '   ' TO CODE:3
  @ 10, 24 GET CODE:3 PICTURE '999'
  READ
  USE OFF
  COUNT ALL FOR CODE:3 = OFF:CODE TO NUM
  IF NUM <> 0
    * ----------------------------------
    * ВВЕДЕН ИСПОЛЬЗУЕМЫЙ КОД ДОЛЖНОСТИ
    * ----------------------------------
    STORE Y TO EXIST
  ELSE
    @ 23, 0 SAY;
    'ОШИБКА: УКАЗАННЫЙ КОД ДОЛЖНОСТИ НЕ ИСПОЛЬЗУЕТСЯ'
  ENDIF
ENDDO
@ 23, 0 SAY;
'
RETURN
```

Введенный код должности помещается в переменную памяти CODE:3.

Командный файл GETCODE4.CMD обеспечивает ввод кода работы, в которой принимает участие сотрудник. И снова допустим только такой код работы, который содержится в файле базы данных WRK.

Введенный код работы помещается в переменную памяти CODE:4.

Как вы, наверно, обратили внимание, три рассмотренных последними командных файла практически идентич-

```
* -----------------------------------------
* КОМАНДНЫЙ ФАЙЛ GETCODE4.CMD
* -----------------------------------------
STORE N TO EXIST
* --------------------------------------------
* ЦИКЛ ВВОДА КОДА РАБОТЫ
* ЦИКЛ ЗАВЕРШАЕТСЯ, ЕСЛИ ВВЕДЕН
* ИСПОЛЬЗУЮЩИЙСЯ КОД РАБОТЫ
* --------------------------------------------
DO WHILE .NOT. EXIST
  STORE '  ' TO CODE:4
  @ 14, 24 GET CODE:4 PICTURE '99'
  READ
  USE WRK
  COUNT ALL FOR CODE:4 = WRK:CODE TO NUM
  IF NUM <> 0
    * ----------------------------------------
    * ВВЕДЕН ИСПОЛЬЗУЕМЫЙ КОД РАБОТЫ
    * ----------------------------------------
    STORE Y TO EXIST
  ELSE
    @ 23, 0 SAY;
    'ОШИБКА: УКАЗАННЫЙ КОД РАБОТЫ НЕ ИСПОЛЬЗУЕТСЯ'
  ENDIF
ENDDO
@ 23, 0 SAY;
'
RETURN
```

ны. Все различия заключаются в том, с какими файлами базы данных и переменными памяти они работают. Несколько различаются сообщения об ошибках, которые могут выводить на экран дисплея эти командные файлы.

Командный файл GETTAX.CMD

После стольких злоключений ввод оклада не представит для вас никаких трудностей.

```
* ----------------------------------
* КОМАНДНЫЙ ФАЙЛ GETTAX.CMD
* ----------------------------------
STORE 0 TO TAX
* ------------------------------
* ВВОД ОКЛАДА СОТРУДНИКА
* ------------------------------
@ 12, 24 GET TAX PICTURE '999'
READ
RETURN
```

Все, что вы видите в командном файле GETTAX.CMD, не должно показаться незнакомым.

Командный файл WRITEEMP.CMD

Цель всей работы, о которой идет речь, состоит в том, чтобы в файл базы данных EMP была внесена корректная запись о сотруднике.

К началу выполнения командного файла WRITEEMP. CMD переменные памяти уже содержат всю информацию, необходимую для внесения записи о сотруднике. Тем не менее у вас может возникнуть желание отказаться от внесения этой записи в файл базы данных. Поэтому было бы не плохо задать вопрос: желаете вы вносить запись или нет? Для вывода на экран дисплея вопроса и получения ответа используется команда @, содержащая SAY- и GET-части. Это значит, что сначала будет выведен текст вопро-

са и курсор остановится сразу за ним. Именно здесь находится область экрана дисплея, в которую будет осуществляться ввод ответа. Введенный вами ответ помещается в переменную памяти REPLY. Если вами введен символ N (или русский Н), то это будет означать отказ от

```
* ----------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ WRITEEMP.CMD
* ВНЕСЕНИЕ ЗАПИСИ О СОТРУДНИКЕ В БАЗУ ДАННЫХ
* ----------------------------------------------------
STORE ' ' TO REPLY
@ 23, 0 SAY 'ВНЕСТИ ЗАПИСЬ В БАЗУ ДАННХ (Д/Н)? ';
GET REPLY PICTURE 'A'
READ
IF REPLY <> 'Н' .AND. REPLY <> 'N'
  * ----------------------------------
  * ВНЕСТИ ЗАПИСЬ В БАЗУ ДАННЫХ
  * ----------------------------------
  USE EMP
  APPEND BLANK
  GOTO BOTTOM
  REPLACE EMP:CODE WITH CODE:1, EMP:NAME WITH NAME;
  DIV:CODE WITH CODE:2, OFF:CODE WITH CODE:3;
  EMP:TAX WITH TAX, WRK:CODE WITH CODE:4
ENDIF
RETURN
```

внесения записи в файл базы данных. В противном случае будут выполнены команды, обеспечивающие дополнение новой записи о сотруднике. Это делается в три шага:

- к файлу базы данных дополняется запись, все поля которой заполнены пробелами (APPEND BLANK);
- затем производится позиционирование на эту запись.

  Это нужно сделать для того, чтобы имелась возможность работы с только что внесенной записью;

- теперь надо заменить поля этой записи, заполненные пробелами, той информацией, которая хранится в переменных памяти.

Сделать это позволяет команда REPLACE. В ней вы указываете имя поля записи, а затем, после слова WITH, задаете имя переменной памяти, значение которой будет помещено в это поле записи. В нашем случае происходит изменение значений шести полей записи файла базы данных EMP.

Обратите внимание! Команда REPLACE размещена в трех строках.

И опять признаком продолжения команды служит точка с запятой.

Заключение к примеру

Наконец-то вы можете оценить, чего стоит обеспечение целостности базы данных, хотя в примере рассматривалось заполнение только одного из четырех файлов, составляющих базу данных. Да и сам пример нельзя считать завершенным. Было бы весьма полезным обеспечить вывод наименований подразделения, должности и работы, соответствующих кодам. Еще бóльшие удобства были бы предоставлены, если бы имелась возможность изменять отдельные введенные данные. Единственно возможное решение этой проблемы, доступное в рамках данного примера, — отказаться от внесения записи в файл базы данных и осуществить полный повтор ввода данных.

Приведенные здесь замечания могут помочь вам при решении своих задач с использованием системы управления реляционной базой данных РЕБУСа.

6.5.5. ВЫБОРКА ИНФОРМАЦИИ ИЗ БАЗЫ ДАННЫХ

Что мы будем искать в базе данных?

Продолжим работу с нашим хорошим знакомым — файлом базы данных EMP — хранилищем записей о со-

трудниках. Задача будет состоять в том, чтобы вывести на экран дисплея информацию о сотруднике. Для выполнения запроса должно быть указано одно из двух:

- код сотрудника;
- фамилия, имя и отчество или несколько первых, а может, и одна буква.

При выводе информации необходимо отобразить на экране дисплея не только коды, содержащиеся в записи о сотруднике, но и соответствующие им наименования.

Теперь остается набраться терпения и посмотреть, как все это делается.

Командный файл **SHOWEMP.CMD**

Командный файл SHOWEMP.CMD начинает решение поставленной задачи.

```
* -----------------------------------
* КОМАНДНЫЙ ФАЙЛ SHOWEMP.CMD
* ВЫВОД ИНФОРМАЦИИ О СОТРУДНИКЕ
* -----------------------------------
SET COLON OFF
SET TALK OFF
SET CONFIRM ON
STORE N TO CANCEL
* -------------------------------------------------------
* ЦИКЛ ВВОДА И ОБРАБОТКИ ЗАПРОСА К БАЗЕ ДАННЫХ
* НА ВЫВОД ИНФОРМАЦИИ О СОТРУДНИКЕ
* ПРИЗНАКОМ ОКОНЧАНИЯ ОБРАБОТКИ ЗАПРОСОВ ЯВЛЯЕТСЯ
* ВВОД ПУСТЫХ СТРОК "КОД СОТРУДНИКА" И "Ф.И.О."
* -------------------------------------------------------
DO WHILE .NOT. CANCEL
  DO GETCDNM
  IF .NOT. CANCEL
    DO FINDEMP
  ENDIF
ENDDO
ERASE
CLEAR
RETURN
```

В нем производится установка параметров системы управления базой данных, а также организуется цикл ввода и выполнения запроса на поиск информации о сотруднике. Как уже говорилось выше, поиск может осуществляться либо по коду сотрудника, либо по фамилии, имени и отчеству. Цикл завершается, если в качестве обоих этих параметров введены «пустые» значения.

Более подробное рассмотрение этого командного файла не имеет смысла. Все приемы работы, использованные в нем, были описаны в предыдущем примере.

Командный файл GETCDNM. CMD

Командный файл GETCDNM.CMD обеспечивает вывод формы для ввода запроса к базе данных и отображения информации. Здесь же происходит ввод запроса на поиск информации о сотруднике.

Для того чтобы можно было производить поиск информации, необходимо дать возможность пользователю в более или менее удобной для него форме указать, что он хочет от компьютера. Именно от пользователя зависит выбор режима поиска: по коду сотрудника или по фамилии, имени и отчеству. А задача разработчика состоит в том, чтобы компьютер мог «понять», каким образом следует вести поиск. Здесь использовано, по-видимому, самое простое решение.

Ввод запроса начинается с предоставления возможности ввести код сотрудника. Если вы введете не «пустой» код сотрудника, то будет производиться поиск по этому параметру. В противном случае вам будет предоставлена возможность ввести фамилию, имя и отчество сотрудника. Повторный «пустой» ввод приведет к окончанию работы.

Переменная памяти логического типа FL:CODE используется для определения параметра, по которому должен производиться поиск. Значение .T. свидетельствует о необходимости поиска по коду сотрудника, а .F. — о поиске по фамилии, имени и отчеству.

```
* -------------------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ GETCDNM.CMD
* ВВОД КОДА СОТРУДНИКА ИЛИ Ф.И.О., ПО КОТОРЫМ
* БУДЕТ ПРОИЗВОДИТЬСЯ ПОИСК В БАЗЕ ДАННЫХ
* -------------------------------------------------------------
ERASE
@ 1, 29 SAY 'СВЕДЕНИЯ О СОТРУДНИКЕ'
@ 3, 0 SAY '                  КОД СОТРУДНИКА: '
@ 5, 0 SAY '      ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО: '
@ 7, 0 SAY '---------------------------------'
@ 9, 0 SAY '               КОД ПОДРАЗДЕЛЕНИЯ: '
@ 10, 0 SAY 'НАИМЕНОВАНИЕ ПОДРАЗДЕЛЕНИЯ: '
@ 12, 0 SAY '                 КОД ДОЛЖНОСТИ: '
@ 13, 0 SAY '                     ДОЛЖНОСТЬ: '
@ 15, 0 SAY '                         ОКЛАД: '
@ 17, 0 SAY '                   КОД РАБОТЫ: '
@ 18, 0 SAY '       НАИМЕНОВАНИЕ РАБОТЫ: '
STORE '   ' TO CODE
STORE '                              ' TO NAME
STORE Y TO FL:CODE
@ 3, 28 GET CODE PICTURE '999'
READ
IF CODE = '
  * ----------------------------------------------------------
  * ПОИСК НЕ БУДЕТ ВЕСТИСЬ ПО КОДУ СОТРУДНИКА
  * ----------------------------------------------------------
  STORE N TO FL:CODE
  @ 5, 28 GET NAME PICTURE;
  'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  READ
  IF NAME = '
    * -----------------
    * ОКОНЧАНИЕ РАБОТЫ
    * -----------------
    STORE Y TO CANCEL
  ENDIF
ENDIF
RETURN
```

Командный файл FINDEMP.CMD

После того как введен запрос, можно приступить к поиску нужной информации.

Первая часть поиска выполняется в файле базы данных EMP. Переменная памяти FL:CODE указывает, какое

```
* ---------------------------------------------
* КОМАНДНЫЙ ФАЙЛ FINDEMP.CMD
* ПОИСК В БАЗЕ ДАННЫХ ЗАПИСИ О СОТРУДНИКЕ ПО
* УКАЗАННОМУ КОДУ ИЛИ Ф.И.О.
* ---------------------------------------------
USE EMP
IF FL:CODE
  * -------------------------
  * ПОИСК ПО КОДУ СОТРУДНИКА
  * -------------------------
  LOCATE FOR EMP:CODE = CODE
ELSE
  * ---------------
  * ПОИСК ПО Ф.И.О.
  * ---------------
  LOCATE FOR EMP:NAME = TRIM(NAME)
ENDIF
IF .NOT. EOF
  * ------------------------------
  * ЗАПИСЬ О СОТРУДНИКЕ НАЙДЕНА
  * ------------------------------
  @ 3, 28 SAY EMP:CODE USING '999'
  @ 5, 28 SAY EMP:NAME USING;
  'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 9, 28 SAY DIV:CODE USING '99'
  @ 12, 28 SAY OFF:CODE USING '999'
  @ 15, 28 SAY EMP:TAX USING '999'
  @ 17, 28 SAY WRK:CODE USING '99'
  DO SHOWREST
  @ 21, 0
ELSE
  @ 21, 0 SAY 'СВЕДЕНИЯ ОТСУТСТВУЮТ'
ENDIF
WAIT
RETURN
```

поле файла базы данных EMP надо использовать при поиске. Для начала необходимо получить доступ к подходящей записи файла базы данных. Это делается посредством команды LOCATE. С ее помощью можно получить доступ к записи, удовлетворяющей условию, указанному в команде следом за словом FOR — EMP:CODE=CODE. Таким образом будет произведена попытка найти запись, в поле EMP:CODE которой содержится значение, совпадающее с кодом сотрудника, введенным в качестве запроса. Поиск сотрудника по фамилии, имени и отчеству обладает одной особенностью, о которой уже говорилось: в запросе фамилия, имя и отчество могут быть указаны не полностью. Что это значит? Например, можно ввести в качестве запроса ПЕ вместо полного ввода фамилии, имени и отчества. Тем не менее при поиске должна быть найдена запись о ПЕТРОВе ПЕТРе ПЕТРОВИЧе.

На этот случай РЕБУС имеет интересное средство — функцию TRIM, которая оставляет от ПЕ и 28 пробелов за ним, введенных в качестве запроса, только ПЕ, отбрасывая все «хвостовые» пробелы. А в команде LOCATE сравнение строк производится с учетом количества символов во второй строке. Такой режим сравнения строк в командах системы управления базой данных устанавливается сразу после запуска РЕБУСа. Это значит, что равными будут признаны

ПЕТРОВ ПЕТР ПЕТРОВИЧ и ПЕ, ПЕТРОВ, ПЕТРОВ П

Итак, предпринята попытка найти нечто подходящее. Но нельзя быть уверенным, что это удалось сделать. Помочь нам может функция EOF. Если подходящая запись была найдена, то значение этой функции будет .F. В противном случае по команде LOCATE были просмотрены все записи файла базы данных и не найдено ни одной подходящей. Команда LOCATE завершит работу, когда, попытавшись считать очередную запись, обнаружит, что ее нет. Это значит, что достигнут конец файла базы данных и функция EOF после этого возвратит значение .T. Именно

такая ситуация возникает в рассматриваемом примере при попытке чтения записи 00007 файла базы данных EMP. Подводя итог, можно сказать, что, если значение функции EOF—.F., то подходящая запись о сотруднике найдена. В противном случае запись не обнаружена. О последней из перечисленных ситуаций следует вывести сообщение на экран дисплея.

С чем вам еще не приходилось сталкиваться? Команда @ 21,0 выполняет только позиционирование курсора в указанное место экрана дисплея. Команда WAIT приостановит выполнение командного файла и выведет сообщение ПАУЗА. Продолжить выполнение командного файла можно, нажав любую клавишу.

Командный файл SHOWREST.CMD

После того как была найдена подходящая запись о сотруднике, остается вывести наименования подразделения, должности и работы, соответствующие кодам, хранящимся в найденной записи о сотруднике.

В процессе решения задачи возникает проблема обработки двух файлов базы данных одновременно. Сначала должны быть совместно обработаны файлы базы данных EMP и DIV для получения наименования подразделения, затем EMP и OFF — для получения наименования должности и, наконец, EMP и WRK — для получения наименования работы. РЕБУС позволяет решить и эту задачу.

До сих пор при работе с файлами базы данных использовалась так называемая первичная область. Ее было достаточно, поскольку не требовалось, чтобы в командах системы управления базой данных обрабатывались поля или записи двух файлов базы данных одновременно. Теперь такая необходимость возникла.

К началу работы командного файла SHOWREST.CMD в первичной области используется файл базы данных EMP, позиционированный на нужную запись. Поэтому

для работы с файлами базы данных DIV, OFF и WRK придется воспользоваться вторичной областью. Первичная и вторичная области не зависят друг от друга. Это значит, что при выполнении, например, поиска в файле базы данных во вторичной области не происходит никаких изменений в позиционировании файла базы данных в первичной области.

Указать системе управления базой данных, что далее надо работать во вторичной области, можно, воспользовавшись командой SELECT SECONDARY. А дальше ничто не мешает обрабатывать во вторичной области файлы базы данных DIV, OFF и WRK.

```
* -----------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ SHOWREST.CMD
* ВЫВОД НАИМЕНОВАНИЙ ПОДРАЗДЕЛЕНИЯ, ДОЛЖНОСТИ, РАБОТЫ
* -----------------------------------------------------
SELECT SECONDARY
* ------------------------------------
* ВЫВОД НАИМЕНОВАНИЯ ПОДРАЗДЕЛЕНИЯ
* ------------------------------------
USE DIV
LOCATE FOR P.DIV:CODE = S.DIV:CODE
@ 10, 28 SAY S.DIV:NAME USING;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
* ---------------
* ВЫВОД ДОЛЖНОСТИ
* ---------------
USE OFF
LOCATE FOR P.OFF:CODE = S.OFF:CODE
@ 13, 28 SAY S.OFF:NAME USING 'XXXXXXXXXXXXXXXXXXXX'
* -------------------------
* ВЫВОД НАИМЕНОВАНИЯ РАБОТЫ
* -------------------------
USE WRK
LOCATE FOR P.WRK:CODE = S.WRK:CODE
@ 18, 28 SAY S.WRK:NAME USING;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
SELECT PRIMARY
RETURN
```

Сначала производится вывод наименования подразделения, в котором работает сотрудник. Решить эту задачу можно с помощью команд LOCATE и @. Особенность их использования связана с именами полей файлов базы данных в первичной и вторичной областях. Используя команду LOCATE, надо найти такую запись из файла базы данных DIV, в поле DIV:CODE которой хранится значение, совпадающее со значением DIV:CODE в записи, на которую позиционирован в первичной области файл базы данных EMP. Приставка S. Означает, что речь идет о поле записи файла базы данных во вторичной области, а P. — в первичной области.

Пользуясь этими нехитрыми приемами, можно найти и вывести всю остальную нужную информацию. Завершив этот процесс, нечего больше делать во вторичной области и можно вернуться к первичной, воспользовавшись командой SELECT PRIMARY.

В чем польза корректных данных?

При обработке файлов базы данных DIV, OFF и WRK не было затрачено ни одной команды на анализ ситуации, при которой в них не была бы найдена запись, содержащая нужный код. Почему же? Да потому, что при вводе записи о сотруднике была проявлена забота о том, чтобы были введены только такие коды подразделения, должности и работы, которые уже хранятся в базе данных. Потратив время на обеспечение ввода корректных данных, удалось сэкономить время при решении задачи обработки базы данных. Так что целостность базы данных может представлять не только теоретический, но и практический интерес.

Естественно, в процессе обработки базы данных не должна нарушаться ее целостность. В этом плане любая мелочь может доставить вам массу неприятностей. Учесть по возможности бóльшую их часть — одна из задач проектирования базы данных.

6.5.6. ПОИСК ИНФОРМАЦИИ О СОТРУДНИКЕ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ

В чем недостаток приведенного примера?

Только что рассмотренный пример поиска информации о сотруднике имеет один недостаток с теоретической точки зрения. При поиске по фамилии, имени и отчеству может возникнуть ситуация, когда в файле базы данных есть несколько подходящих записей. Например, если вы попытаетесь найти сотрудников, фамилия которых начинается с ДУ, то обнаружена будет только запись о Дубовой Татьяне Петровне. Что касается Дуровой Юлии Сергеевны, то запись о ней найдена не будет. Проблема возникает из-за того, что командные файлы обеспечивают поиск только первой подходящей записи о сотруднике, а затем происходит переход к вводу нового запроса.

Воспользуемся знанием теории

Решим задачу следующим образом. Сформируем рабочий файл базы данных, содержащий всю необходимую для обработки запроса информацию: код сотрудника; фамилию, имя, отчество; код подразделения; наименование подразделения; код должности; наименование должности; оклад; код работы; наименование работы. Что для этого требуется сделать?

●●● На первом шаге производится операция соединения таблицы СОТРУДНИКИ с результатом выполнения операции проекции, примененной к таблице ПОДРАЗДЕЛЕНИЯ по столбцам «Код подразделения» и «Наименование подразделения». В результате к таблице СОТРУДНИКИ добавится столбец «Наименование подразделения».

Таким образом, в результирующей таблице хранится не только код подразделения, в котором работает сотрудник, но и наименование соответствующего подразделения.

●●● Вторым шагом будет осуществление операции соединения таблицы, полученной в результате выполнения первого шага, с таблицей ДОЛЖНОСТИ.

●●● На третьем шаге производится соединение таблицы, полученной на втором шаге, с результатом выполнения операции проекции применительно к таблице РАБОТЫ по столбцам «Код работы» и «Наименование работы». В результате будет получена таблица, содержащая всю нужную информацию.

Реализация в РЕБУСе

Когда стало очевидным, что требуется сделать, остается определить те средства системы управления базой данных, которые могут выполнить необходимые действия.

```
* ----------------------------------------
* КОМАНДНЫЙ ФАЙЛ EMPASK.CMD
* ОБСЛУЖИВАНИЕ ЗАПРОСОВ О СОТРУДНИКАХ
* ----------------------------------------
ERASE
SET TALK OFF
SET COLON OFF
SET CONFIRM ON
@ 1, 0 SAY 'ФОРМИРУЕТСЯ РАБОЧИЙ ФАЙЛ БАЗЫ ДАННЫХ'
@ 3, 0 SAY 'ПОДОЖДИТЕ ПОЖАЛУЙСТА'
DO CREMPEXT
STORE N TO CANCEL
DO WHILE .NOT. CANCEL
  DO GETCDNM
  IF .NOT. CANCEL
    DO FIEMPEXT
  ENDIF
ENDDO
ERASE
CLEAR
DELETE FILE EMPEXT
RETURN
```

Командный файл EMPASK.CMD организует взаимодействие всех остальных командных файлов, участвующих в решении задачи. По сути требуется сделать следующее:

- сформировать рабочий файл базы данных EMPEXT, в котором затем будет производиться поиск;
- обеспечить ввод и обработку запроса на поиск информации о сотруднике;
- завершить работу, удалив использовавшийся файл базы данных (команда DELETE FILE EMPEXT).

Как вы помните, формирование рабочего файла для

```
* ---------------------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ CREMPEXT.CMD
* ФОРМИРОВАНИЕ ФАЙЛА БАЗЫ ДАННЫХ ДЛЯ ОБСЛУЖИВАНИЯ
* ЗАПРОСОВ О СОТРУДНИКАХ
* ---------------------------------------------------------------
USE EMP
SELECT SECONDARY
USE DIV
SELECT PRIMARY
JOIN TO TMP1 FOR DIV:CODE = S.DIV:CODE FIELDS EMP:CODE,;
EMP:NAME,DIV:CODE,S.DIV:NAME,OFF:CODE,EMP:TAX,WRK:CODE
USE TMP1
SELECT SECONDARY
USE OFF
SELECT PRIMARY
JOIN TO TMP2 FOR OFF:CODE = S.OFF:CODE FIELDS EMP:CODE,;
EMP:NAME,DIV:CODE,DIV:NAME,OFF:CODE,S.OFF:NAME,EMP:TAX,;
WRK:CODE
USE TMP2
DELETE FILE TMP1
SELECT SECONDARY
USE WRK
SELECT PRIMARY
JOIN TO EMPEXT FOR WRK:CODE = S.WRK:CODE FIELDS EMP:CODE,;
EMP:NAME,DIV:CODE,DIV:NAME,OFF:CODE,OFF:NAME,EMP:TAX,;
WRK:CODE,;
S.WRK:NAME
USE EMPEXT
DELETE FILE TMP2
RETURN
```

обслуживания запросов о сотрудниках требует выполнения операций соединения и проекции. В РЕБУСе для этих целей существует сверхмощное средство — команда JOIN. В зависимости от того, как вы ею воспользуетесь, можно выполнить операции прямого произведения или соединения. Причем одновременно может быть выполнена и операция проекции.

Команда JOIN использована в командном файле CREMPEXT.CMD для соединения файлов базы данных с попутным выполнением проекции. Эта команда обрабатывает файлы баз данных из первичной и вторичной областей и состоит из шести элементов:

- слова TO, означающего, что далее будет указано имя файла базы данных, в который будет помещен результат;
- имени результирующего файла базы данных;
- слова FOR, говорящего о том, что далее будет следовать некоторое условие;
- условия, при выполнении которого запись помещается в результирующий файл базы данных;
- слова FIELDS;

```
* ---------------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ FIEMPEXT.CMD
* ПОИСК ЗАПИСИ О СОТРУДНИКЕ ПО
* УКАЗАННОМУ КОДУ ИЛИ Ф.И.О.
* ---------------------------------------------------------
IF FL:CODE
  LOCATE FOR EMP:CODE = CODE
 ELSE
  LOCATE FOR EMP:NAME = TRIM(NAME)
ENDIF
IF .NOT. EOF
  DO SHEMPEXT
ELSE
  @ 21, 0 SAY 'СВЕДЕНИЯ ОТСУТСТВУЮТ'
  WAIT
ENDIF
RETURN
```

◆ списка полей файлов базы данных из первичной и вторичной области, образующих структуру результирующего файла базы данных.

Используемые для хранения промежуточных результатов файлы базы данных TMP1 и TMP2 удаляются по мере того, как получаются новые результаты. В конце концов будет получен файл базы данных EMPEXT, ради которого все это и делалось.

Командный файл GETCDNM.CMD используется в том виде, в котором он приведен в предыдущем примере.

Действия, выполняемые командным файлом FIEMPEXT. CMD, едва ли могут вызвать у вас непонимание.

Ничего особенно сложного не делает и командный файл SHEMPEXT.CMD. Маленькая хитрость связана с командой CONTINUE. Эта команда позволяет повторить

```
* ------------------------------------------------------------
* КОМАНДНЫЙ ФАЙЛ SHEMPEXT.CMD
* ОТОБРАЖЕНИЕ ПОДХОДЯЩИХ ЗАПИСЕЙ О СОТРУДНИКАХ
* ------------------------------------------------------------
DO WHILE .NOT. EOF
  @ 3, 28 SAY EMP:CODE USING '999'
  @ 5, 28 SAY EMP:NAME USING;
  'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 9, 28 SAY DIV:CODE USING '99'
  @ 10, 28 SAY DIV:NAME USING;
  'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 12, 28 SAY OFF:CODE USING '999'
  @ 13, 28 SAY OFF:NAME USING 'XXXXXXXXXXXXXXXXXXXX'
  @ 15, 28 SAY EMP:TAX USING '999'
  @ 17, 28 SAY WRK:CODE USING '99'
  @ 18, 28 SAY WRK:NAME USING;
  'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 21, 0
  WAIT
  @ 23, 0 SAY
  CONTINUE
ENDDO
RETURN
```

поисковую операцию, определенную последней встретившейся перед CONTINUE командой LOCATE. Таким образом, команда LOCATE находит первую подходящую запись, а циклическое использование команды CONTINUE позволяет найти все остальные подходящие записи, пока не будет достигнут конец файла.

Сравним полученные реализации

●●● Первая реализация требует больше времени непосредственно на поиск информации. Это связано с необходимостью ведения поиска в четырех файлах базы данных во время обработки запроса.

●●● Во второй реализации выполняется достаточно длительная подготовка к работе, обусловленная формированием рабочего файла базы данных. Впоследствии это окупается за счет более быстрой по сравнению с первой реализацией обработки запроса. Она может быть полезна в том случае, когда требуется удовлетворить сразу большое количество запросов о сотрудниках. Другим преимуществом второй реализации является строгое описание выполняемых преобразований, способствующее большей четкости в понимании решаемой вами задачи и определении способа получения желаемого результата.

6.6. ОБЩАЯ СПРАВКА О РЕБУСе

6.6.1. ОСНОВНЫЕ ПОНЯТИЯ

К этому моменту вы ознакомились с процессом создания базы данных и возможностями, которые предоставляет система управления базой данных (СУБД) для организации автоматической обработки базы данных с помощью командных файлов. Тем не менее даже те свойства СУБД,

которые были описаны, нельзя считать рассмотренными всесторонне и в полном объеме. В связи с этим нижеследующий материал можно расценивать как попытку систематического описания элементов системы управления базой данных СУБД РЕБУСа и пользоваться им в качестве справочного при работе с РЕБУСом; однако это не означает, что в будущем не возникнет необходимости в использовании документации [17].

Термины и определения

Переменная — это либо переменная базы данных, либо переменная памяти. Переменная базы данных — это любое поле записи базы данных. Переменная памяти — это величина, хранящаяся в оперативной памяти компьютера.

Константа — это величина, имеющая неизменное значение независимо от базы данных и переменных памяти.

Символьные константы (совокупность символов) заключаются в апострофы ('), кавычки (") или квадратные скобки ([ ]). Если символьная константа содержит один из этих ограничителей, то она должна быть ограничена другим. Логические константы представляются как символы "T" (или "Y") для обозначения истинного (TRUE) и "F" (или "N") для обозначения ложного (FALSE).

Имя переменной — это последовательность от 1 до 10 букв, цифр и двоеточий (:), начинающаяся с буквы и заканчивающаяся буквой или цифрой.

Тип данных — это один из следующих типов, допустимых в РЕБУСе: символьный (C), числовой (N), логический (L).

Длина поля данных — это число позиций для представления данных, которое должно находиться в пределах от 1 до 254 (для числовых данных учитывается и позиция, занимаемая точкой в десятичном числе).

Выражение — это допустимое в РЕБУСе описание вычисления значения, состоящее из переменных, констант, обращений к функциям и операций.

Текущий файл — это тот файл базы данных, который был определен последней командой USE.

Текущая запись — это та запись, на которую осуществлено позиционирование в текущий момент времени. Записи могут иметь номера от 0 до 65535.

Условные обозначения терминов РЕБУСа

Перед тем как перейти к дальнейшему изучению элементов РЕБУСа, ознакомьтесь с некоторыми обобщенными обозначениями и соглашениями:

<команда> — любая допустимая в РЕБУСе команда;

<строка> — любая символьная строка;

<выражение> — выражение;

<спис.выр.> — список выражений, разделенных запятыми;

<поле> — имя поля записи;

<список> — список имен полей записей, разделенных запятыми;

<файл> — имя файла;

<форм.файл> — имя форматного файла;

<инд.файл> — имя индексного файла;

<ключ> — имя поля записи, по которому будет производиться индексирование;

<переменная> — имя переменной памяти или поля записи;

<перем.пам.> — имя переменной памяти;

<список перем.пам.> — список имен переменных памяти, разделенных запятыми;

<N> — константа;

<диапазон> — область действия команды, определяемая следующими значениями:

ALL — все записи в базе данных.

После позиционирования на начало базы данных команда обрабатывает все записи базы данных;

NEXT N — следующие N записей базы данных;

RECORD N — N-я запись базы данных;

Таблица 6.6

Типы файлов РЕБУСа

Тип файла	Название файла	Назначение
.DBF	Файл базы данных	Хранение описания структуры и записей базы данных
.MEM	Файл содержимого памяти	Сохранение для дальнейшего использования на диске информации, содержащейся в переменных памяти
.CMD	Командный файл	Хранение последовательности команд СУБД в целях их автоматического выполнения
.FRM	Файл отчетов	Строится во время диалога при выполнении команды REPORT и содержит описание формы отчета
.TXT	Текстовые файлы	Хранение текстовой информации, которая возникает или требуется при выполнении команд СУБД
.NDX	Индексный файл	Содержит указатели на записи файла базы данных
.FMT	Форматный файл	@ Хранение последовательности команд и * , которая используется при задании этого файла в команде SET FORMAT TO
.SDF	Системный файл	Содержит текстовые данные в формате, принятом в операционной системе SCP
.BAK	Предыдущая версия файла	Предыдущия версия файла может сохраняться на диске при выполнении некоторых команд СУБД

FOR <выражение> — любая запись базы данных, если логическое выражение истинно;

WHILE <выражение> — все записи базы данных, пока логическое выражение истинно.

Обозначения, не указанные в этом списке, будут пояснены по мере их появления.

Элементы, заключенные в квадратные скобки ([ ]), указывать необязательно и они могут быть опущены.

Файлы РЕБУСа

Все файлы, создаваемые в процессе работы РЕБУСа, являются стандартными файлами операционной системы SCP с именем, содержащим до восьми символов, и с типом из трех символов. Описание используемых в РЕБУСе типов файлов приведено в табл. 6.6.

Ограничения

Максимальное количество:	
полей в записи	32
символов в записи	1000
записей базы данных	65535
символов в строке	254
разрядов в числовом поле	10
Число:	
наибольшее	$1.0*10^{-63}$
наименьшее	$1.8*10^{63}$
Максимальное количество:	
переменных памяти	64
символов в команде	254
выражений в команде SUM	5
символов	
в ключе	99
» заголовке отчета	254
отложенных команд @ с GET частью	64
одновременно открытых файлов	16

Макроподстановки

Каждый раз, когда в команде встречается символ &, за которым следует имя переменной памяти, имеющей сим-

вольный тип, РЕБУС заменяет его значением этой переменной.Если вслед за & и именем переменной памяти необходимо поставить алфавитно-цифровой символ, то после имени переменной должна следовать точка (.), которая будет удалена во время выполнения макроподстановки.

Пример.

```
STORE "B" TO DRIVE
RESET & DRIVE
USE &DRIVE.:EMP
```

6.6.2. ФУНКЦИИ

Функции вы можете использовать в любом выражении. Тип функции определяется типом получаемого результата. В табл. 6.7 приведены функции РЕБУСа, классифицированные по типу результатов.

6.6.3. ОПЕРАЦИИ

Операции используются для формирования выражений и делятся на арифметические, логические, строковые, операции сравнения. В табл. 6.8—6.11 представлены операции в порядке убывания их приоритетов.

Арифметические операции (табл 6.8) предназначены для выполнения традиционных арифметических действий.

Таблица 6.8

Арифметические операции

Операция	Описание	Пример использования
+, —	Знак числа	
*, /	Умножение Деление	A * B A/B
+, —	Сложение Вычитание	A + B A — B

Таблица 6.7

Функции СУБД РЕБУСа

Функция	Описание	Пример	
		Использования	Результат
	Числовые функции		
INT (<числовое выр.>)	Возвращает значение числового выражения и отбрасывает дробную часть	INT (3.14)	3
#	Возвращает целое число, соответствующее номеру текущей записи базы данных	#	4 (текущая — 4-я запись БД)
VAL (<строка>)	Возвращает целое число, сформированное из символьной строки. Если строка начинается с цифровых символов и далее содержит нецифровые символы, то для формирования используются только цифровые символы	VAL ("1988 год")	1988
LEN (<строка>)	Возвращает целое число, равное числу символов, содержащихся в строке	LEN ("1988 год")	8
@ (<строка 1>, <строка 2>)	Возвращает целое число, значение которого указывает позиция в <строке 2>, начиная с которой <строка 1> совпадает со <строкой 2>	@ ("88", "1988 год")	3

RANK (<строка>)	Возвращает код первого символа <строки>	RANK ("1988 год")	49
	Символьные функции		
STR (числовое выр.>, <длина>, [<дес>])	Возвращает строку символов, содержащую <длина> символов, полученную в результате преобразования значения числового выражения; <дес> может задавать количество цифр после точки в десятичном числе	STR (3.14, 5,1)	3.1
¤ (<строковое выр.>, <начало>, <длина>)	Возвращает строку длиной <длина>, содержащую символы из строки, полученной в результате вычисления символьного выражения, начиная с позиции <начало> и длиной <длина> символов. Если <длина> больше длины результата вычисления символьного выражения или начало <длина> выходит за его границу, то в качестве результата будут возвращаться только имеющиеся символы	¤ ("1988год", 5,5)	год
! (<символьное выражение>)	Возвращает строку, полученную в результате вычисления символьного выражения, в которой все строчные буквы заменены прописными	! ("ГОД")	GOD
CHR (<числовое выражение >)	Возвращает символ, код которого соответствует полученному значению числового выражения	CHR (48)	Ø

Функция	Описание	Пример	
		Использования	Результат
DATE ( )	Возвращает строку, содержащую текущую дату в формате дд/мм/гг (дд — день, мм — месяц, гг — год)	DATE ( )	05/01/88
TYPE (<выражение>)	Возвращает односимвольную строку, значение которой зависит от типа выражения: C — символьное, N — числовое, L — логическое	TYPE (3.14*3.14)	N
TRIM (<строка>)	Возвращает строку, полученную из <строка> удалением концевых пробелов	TRIM ("1988␣␣")	1988
	Логические функции		
*	Возвращает значение истины, если текущая запись была помечена на удаление, иначе — ложь		
EOF	Возвращает значение истины при достижении конца текущего файла базы данных, иначе ложь		
FILE (<имя файла>)	Возвращает значение истины, если файл с указанным именем существует, иначе — ложь. Имя файла задается как символьная константа		

Логические операции (табл. 6.9) служат для выполнения действий, определяемых в булевой алгебре. Эти операции используют значения, имеющие смысл истина/ложь (.T./.F.). Логические операции задаются в виде таблиц истинности. Эти таблицы ставят в соответствие всем возможным значениям аргументов значение результата операции.

Таблица 6.9

Логические операции

Операция	Описание			Пример
	арг. 1	арг. 2	Результат	
.NOT. (отрицание)	.T. .F.		.F. .T.	.NOT. A
.AND. (логическое И)	.F .F. .T. .T.	.F. .T. .F. .T.	.F. .F. .F. .T.	A .AND. B
.OR. (логическое ИЛИ)	.F. .F. .T. .T.	.F. .T. .F. .T.	.F. .T. .T. .T.	A .OR. B

В РЕБУСе определены две операции конкатенации (сцепления) строк. В результате выполнения этих операций из двух строк формируется третья, представляющая собой строку, в которой за первой строкой следует вторая. Строковые операции описаны в табл. 6.10.

Операции сравнения (табл. 6.11) выполняют сравнение двух значений, в результате чего получается логическое значение .T., если указанное отношение имеет место, и .F. в противном случае.

При рассмотрении правил формирования выражений всегда следует обращать особое внимание на порядок выполнения операций, т. е. на их приоритет. В РЕБУСе принят следующий порядок:

- вычисление значения функций;

Таблица 6.10

Строковые операции

Операция	Описание	Пример	
		Использование	Результат
+	Конкатенация	“ 1988“ + „ год“	„_1988_год“
—	Конкатенация с удалением пробелов	„ 1988“ + “ год“	„_1988 год“

- выполнение арифметических или строковых операций в соответствии с их приоритетами;
- выполнение операций сравнения в соответствии с их приоритетами;
- выполнение логических операций в соответствии с их приоритетами.

Операции одного приоритета осуществляются слева направо. Для изменения порядка выполнения операций можно пользоваться круглыми скобками. Порядок выполнения операций в скобках соответствует приведенному выше и приоритетам операций.

Таблица 6.11

Операции сравнения

Операции	Описание	Пример
=	Равно	A=B
<>	Не равно	A<>B
<	Меньше	A<B
>	Больше	A>B
<=	Не больше	A<=B
>=	Не меньше	A>=B

Параметр	Значение параметра	Описание
		и далее база данных используется как неиндексированная
MARGIN TO	N	Позволяет установить отступ в N пробелов от левого края при выводе на печать. Значение N должно находиться в пределах от 1 до 254

Примечание. Чертой внизу обозначены значения параметров, устанавливаемые при запуске РЕБУСа.

6.6.4. КОМАНДЫ

РЕБУС содержит большое число команд. Попытаемся упорядочить процесс их изучения. Будем рассматривать команды СУБД в соответствии с их функциональным значением, разделив их на девять групп (табл. 6.12):

- команды создания файлов РЕБУСа;
- команды добавления данных;
- команды просмотра и редактирования данных;
- команды позиционирования в базе данных;
- команды работы с файлами РЕБУСа;
- команды работы с переменными;
- команды организации командных файлов;
- прочие команды;
- команда установки параметров СУБД (табл. 6.13).

6.6.5. ПОЛНОЭКРАННЫЙ РЕЖИМ РАБОТЫ

Ряд команд СУБД РЕБУС использует так называемый полноэкранный (FULL SCREEN) режим работы. В этом

Таблица 6.12

Команды РЕБУСа

Команда	Описание
Команды создания файлов РЕБУСа	
CREATE [< файл >]	Создание новой структуры базы данных, предусмотрена возможность дальнейшего ввода данных в соответствии с вводимой структурой
COPY [STRUCTURE TO] < файл >	Производит копирование базы данных в < файл >
[<диапазон>] [FIELD <список>] [FOR<выражение>][SDF] [DELIMITED WITH <ограничитель>]	STRUCTURE указывает на необходимость копирования только структуры данных FIELD <список> — копирование только указанных полей SDE — копирование данных без структуры в файл в системном формате DELIMITED — определяет, каким образом будут разделяться поля данных в базе данных
MODIFY STRUCTURE	Производит изменение структуры базы данных с уничтожением всех ее записей
MODIFY COMMAND [<файл>]	Внесение изменений в командный или любой другой текстовой файл
REPORT [FROM <файл отчета>]	Генерация отчетов с использованием данных из базы данных
[<диапазон>] [TO PRINT]	TO PRINT — вывод на печать параллельно с выводом на дисплей

[FOR <выражение>] [PLAIN]	PLAIN — подавление вывода номеров страниц, и текущей даты вверху каждого листа
	Формирование формата отчета происходит в диалоге с пользователем. Отчет может содержать до 24 граф
SAVE TO <файл> [ALL LIKE <образец> / ALL EXEPT <образец>]	Сохранение всех или указанных переменных памяти LIKE задает сохранение всех переменных имена которых соответствуют <образцу> EXEPT задает сохранение всех переменных за исключением тех, которые соответствуют <образцу> <образец> — одно или несколько имен переменных Использование в образце символа * соответствует последовательности любых символов, а ?— любому символу
INDEX ON <выражение> TO <инд. файл>	Индексирование базы данных, где <выражение> вычисляет ключ, а <инд. файл> задает имя индексного файла
REINDEX	Обновление индексных файлов, которые автоматически не обновлялись во время модификации

Команда	Описание
JOIN TO <файл> FOR <выражение> [FIELDS <список полей>]	Объединяет две базы данных и формирует третью базу данных. Объединяются первичная и вторичная базы данных. Команде JOIN должна предшествовать команда SELECT PRIMARY. Команда JOIN позиционирует первичную базу данных на первую запись и вычисляет <выражение> для каждой записи вторичной базы данных. Если <выражение> истинно, то в <файл> добавляется запись. По достижении конца вторичного файла происходят переход ко второй записи первичной базы данных, позиционирование на начало вторичной базы данных и процесс повторяется
TOTAL ON <ключ> TO <файл> [FIELDS <список>] [FOR <выр>]	Выбор числовых полей базы данных, но суммирование и запись в другую базу данных. Текущая база данных должна быть индексирована по ключу

Команды добавления данных

	Добавление данных в базу:
APPEND	у пользователя запрашиваются данные в соответствии со структурой записи базы данных
APPEND BLANK	к текущему файлу добавляется запись, заполненная пробелами
APPEND FROM <файл> [FOR <выражение>]	добавляемые записи берутся из <файла>;

[SDF] [DELIMITED] WITH <ограничитель>]	SDF задает, что файл в системном формате; DELIMITED задает, что записи в файле ограничены так, как это принято в некоторых языках программирования; если SDF и DELIMITED отсутствуют, то <файл> считается файлом базы данных
CREATE	Команда была рассмотрена выше
INSERT [BEFORE] [BLAHK]	Вставляет запись в базу данных BEFORE — вставка перед текущей записью, в противном случае — после BLANK — вставка пустой записи

Команды просмотра и редактирования данных

CHANGE [<диапазон>] FIELD <список> [FOR <выражение>]	Позволяет ввести новые, модифицировать и просматривать имеющиеся данные; <список> задает поля базы данных, которые будут обрабатываться командой, и последовательность их обработки. По исчерпании списка полей происходит переход к следующей записи. Если <диапазон> не указан, то берется текущая запись
BROWSE [FIELD <список>]	Просмотр и редактирование данных в полноэкранном режиме
DELETE [<диапазон>] [FOR <выражение>]	Пометка на удаление всех записей из <диапазона>, удовлетворяющих <выражению>
DELETE FILE <файл>	Удаление <файла> с диска
EDIT [N]	Выборочное редактирование полей в записях базы данных, N — номер записи базы данных
PACK	Физическое удаление из базы данных всех записей, которые были помечены на удаление

Команда	Описание
RECALL [<диапазон>] [FOR [<выражение>]	Снимает пометки на удаление, сделанные командной DELETE
REPLACE [<диапазон>] <поле 1>WITH <выражение 1> [, <поле 2> WITH <выражение 2>...] [FOR <выражение>]	Замена содержимого полей записей базы данных. Если диапазон не указан, то используется только текущая запись базы данных
UPDATE FROM <файл> ON ключ [ADD <список полей>] [REPLACE <список полей>]	Просмотр базы данных и изменение ее содержимого с использованием данных из файла. Данные либо суммируются (ADD), либо заменяются (REPLACE). Текущая база данных должна быть отсортирована или индексирована по ключу
@ <строка>, <позиция> [SAY <выражение>] [USING <формат>] [GET <переменная>] [PICTURE <формат>]]	Форматированный ввод/вывод информации в полноэкранном режиме
READ	Отображение, просмотр, ввод и редактирование информации в полноэкранном режиме
CLEAR [GET[S]]	[GET[S] — сброс отложенных фраз GET команды без @ сброса экрана. Если не указано GET [S], то происходит переустановка СУБД, т. е. закрываются все базы данных, сбрасываются, переменные памяти
а) DISPLAY [<диапазон>] [FOR <выражение>] [OFF] [FIELD <спис. полей>]] б) DISPLAY STRUCTURE	Отображение информации о состоянии базы данных (а, б), переменных памяти (в), диска (г), об используемых базах данных, индексных файлах, дате и т. п. (д)

в) DISPLAY MEMORY г) DISPLAY FILES [ON <диск>] [LIKE <образец>] д) DISPLAY STATUS	
LIST <ключевые слова>	Аналогична команда DISPLAY за исключением того, что область ее действия — все записи базы данных и она не переходит в ожидание после вывода очередных 15 записей
REPORT	Команда была рассмотрена выше
SUM <числ. выражение 1>, [, числ. выражение 2>...] [TO <спис, перем. парам.>)] [<диапазон>] [FOR <выражение>]	Суммирует значения, полученные в результате вычисления <числ. выражений> в пределах <диапазона>. Если <диапазон> не указан, а также отсутствует часть FOR, то обрабатываются все записи базы данных. Может использоваться одновременно не более пяти выражений. Если используется часть TO, то можно сохранять результаты суммирования в переменной памяти
? <спис. выр.>	Отображает значение выражения или списка выражений. Она может использовать переменные памяти, константы, функции и поля записей базы данных. Команда? без параметров дает строку пробелов
?? <спис. выр.>	Команда ?? аналогична за исключением того, что она не переходит на новую строку перед выводом результата

Команда	Описание
Команды позиционирования в базе данных	
CONTINUE	Используются с командой LOCATE. Эти команды могут быть разделены другими командами
FIND <ключ>	Поиск в индексированной базе данных первой записи, удовлетворяющей <ключу>
GOTO (GO) а) GOTO RECORD <N> б) GOTO TOP в) GOTO BOTTOM г) <N> д) GOTO <перем. пам.>	Переход к записи базы данных с номером <N> (а, г), к первой записи базы данных (а), к последней записи базы данных (в), к записи, номер которой содержится в <перем. пам.>
LOCATE [<диапазон>] [FOR <выражение>]	Поиск в базе данных записи, поля которой удовлетворяют <выражению>
SKIP [+/–] <выражение>	Переход вперед (+) или назад (–) на число записей, определяемое <выражением>; SKIP просто означает переход к следующей записи базы данных
Команды работы с файлами РЕБУСа	
APPEND	Команда была рассмотрена выше
COPY	То же
DELETE	"
RENAME <старое имя файла> TO <новое имя файла>	Изменение имени файла, хранящегося на диске. По умолчанию — тип файла DBF

RESTORE FROM <файл> [ADDITIVE]	Восстановление переменных памяти из <файла> на диске. Если не указано ADDITIVE, то все переменные памяти, существовавшие до ввода команды RESTORE, будут уничтожены
SELECT [PRIMARY / SECONDARY]	Позволяет выбрать для дальнейшей работы первичную или вторичную базу данных. Таким образом, имеется возможность работы с двумя базами данных
SORT ON <поле> TO <файл> [ASCENDING / DESCENDING]	Сортировка базы данных по указанному полю с сохранением результата в базе данных с именем <файл> в возврастающем (ASCENDING) или убывающем (DESCENDING) порядке. Команда едва ли может принести большую пользу, так как она неприменима для сортировки строк символов русского алфавита
USE [<файл>]	Указывает, какая уже существующая база данных будет использоваться
USE <файл> INDEX <инд. файл 1> [..., <инд. файл 7>]	Кроме того, можно указать от одного до семи имеющихся индексных файлов в целях их обновления при модификации базы данных

Команды работы с переменными

ACCEPT [<строка>] TO <перем. пам.>	Ввод строки символов в переменную память. Если указана <строка>, то она будет выведена в качестве подсказки перед вводом данных
COUNT [<диапазон>] [FOR <выражение>] [TO <перем. пам.>]	Подсчет числа записей в базе данных
DISPLAY	Команда была рассмотрена выше

Команда	Описание
INPUT [<строка>] TO <перем. пам.>	Ввод некоторых значений в переменные памяти. Тип <перем. пам.> определяется по типу введенных данных и может быть проанализирован с помощью функции TYPE. Строка символов при вводе должна заключаться в кавычки. Если указана <строка>, то она будет выведена в качестве подсказки перед вводом данных
RELEASE <список перем. пам.>	Освобождает указанные в списке переменные памяти
RELEASE ALL [LIKE <образец>] [EXEPT <образец>]	Без частей LIKE и EXEPT освобождает все переменные памяти. Части LIKE и EXEPT используются аналогично команде SAVE (см. выше)
RESTORE	Команда была рассмотрена выше
SAVE	То же
STORE <выражение> TO <перем. пам.>	Вычисление значения <выражение> и запись его в <перем. пам.>
SUM	Команда была рассмотрена выше
WAIT [TO <перем. пам.>]	Приостановка работы СУБД и ожидание ввода с клавиатуры любого символа. Если имеется часть TO, то введенный символ будет записан в <перем. пам.>. Если введен управляющий символ, то в <перем. пам.> будет записан пробел

Команды организации командных файлов	
CANCEL	Прекращение выполнения командного файла и переход к прямому выполнению команд
DO <файл>	Начинается выполнение команд, находящихся в <файле>
DO WHILE <выражение> <команды>	Пока <выражение> истинно, обрабатываются <команды>. По достижении ложности <выражения> выполняется команда, следующая за командой ENDDO
DO CASE CASE <выражение 1> <команды> . . . CASE <выражение N> <команды> [OTHERWISE <команды>] ENDCASE	Выполняются <команды>, следующие за первым истинным <выражением>. Если такового не обнаружено и указана часть OTHERWISE, то выполняются <команды>, расположенные за ней. После выполнения команды DO CASE будет выполняться команда, следующая за командой ENDCASE
IF <выражение> <команды> [ELSE <команды>] ENDIF	Если <выражение> истинно, то будут выполняться <команды>, следующие за командой IF и <выражением>. Если <выражение> ложно и имеется команда ELSE, то будут выполнены <команды>, следующие за ней. После выполнения команды IF будет выполняться команда, следующая за ENDIF
LOOP	Используется в команде DO WHILE для пропуска команд, следующих в ней за командой LOOP, и приводит к выполнению команды DO WHILE
MODIFY COMMAND	Команда была рассмотрена выше

Команда	Описание
RETURN	Используется в командном файле для возвращения управления вызвавшему его командному файлу или для перехода к прямому выполнению команд
TEXT <любые символы> ENDTEXT	Вывод текстовой информации на экран
WAIT	Команда была рассмотрена выше
Прочие команды	
EJECT	Производится прогон бумаги на печатающем устройстве до конца листа. При использовании команд @ EJECT устанавливает начало листа (т. е. О-я строка и О-я позиция в строке)
ERASE	Очистка экрана и установка курсора в верхний левый угол экрана. При использовании команд @ по команде ERASE производится сброс всех отложенных GET-частей
HELP [<ключевое слово>]	Получение справочной информации о РЕБУСе во время работы с СУБД. Для возможности пользоваться этой командой требуется, чтобы на диске с РЕБУСом находился файл REBUSMSG. IXT. Информацию о возможных режимах работы команды можно получить, введя HELP (или HELP REBUS)

NOTE <любые символы> * <любые символы>	Предназначены для внесения комментариев в командные файлы
QUIT [TO <список программ>]	Окончание работы с СУБД и передача управления операционной системе SCP. Если указана часть TO, то будут выполнены программы, перечисленные в <списке программ>
REMARK <любые символы>	Вывод некоторого текста на устройство вывода
RESET [<дисковод>]	Устанавливает дисковую систему в начальное состояние (сброс дисковой системы). Это необходимо делать всякий раз, когда производится смена диска, так как иначе невозможно произвести запись информации на вновь установленный диск

Таблица 6.13

Команда установки параметров системы SET

Параметр	Значение параметра	Описание
ECHO	ON	Команды, выполняющиеся в командном файле, отображаются на экране
	OFF	Не отображаются
STEP	ON	Остановка выполнения командного файла после каждой команды
	OFF	Остановки нет
TALK	ON	Результаты работы команды отображаются на экране
	OFF	Не отображаются
PRINT	ON	Вывод на печатающее устройство
	OFF	Не производится
CONSOLE	ON	Вывод на дисплей
	OFF	Не производится
ALTERNATE	ON	Вывод в файл на диске
	OFF	Не производится
SCREEN	ON	Разрешены полноэкранные операции
	OFF	Не разрешены
LINKAGE	ON	Позиционирование производится в первичной и вторичной базах данных
	OFF	Независимое позиционирование
COLON	ON	Поля базы данных ограничиваются двоеточиями
	OFF	Границы полей не выводятся
BELL	ON	Звуковой сигнал при вводе неверных данных или нарушение границ полей (Роботрон 1715 не обеспечивает возможности подачи звукового сигнала)

Параметр	Значение параметра	Описание
	OFF	Не подается
ESCAPE	ON	Нажатие клавиши **ESC** приводит к окончанию выполнения командного файла
	OFF	Действие клавиши **ESC** отменено
EXACT	ON	Символьные строки должны сравниваться полностью
	OFF	Символьные строки сравниваются по длине второй строки (истинно "12345" = "123")
INTENSITY	ON	Разрешено использование видеоатрибутов
	OFF	Не разрешено
DEBUG	ON	Вывод, установленный управляющими параметрами ECHO и STEP, посылается на печатающее устройство
	OFF	Вывода на печатающее устройство нет
CARRY	ON	Данные из предыдущей записи пересылаются в следующую запись (например, в команде APPEND)
	OFF	Пересылка отсутствует
CONFIRM	ON	Нет перехода к следующему полю при полноэкранном редактировании без нажатия клавиши ЕТ
	OFF	Переход к следующему полю, если текущее поле заполнено
EJECT	ON	Команда REPORT пропускает страницу перед выводом отчета
	OFF	Пропуска страницы нет
RAW	ON	Одиночные пробелы не вставляются между полями записи, если команды DISPLAY и LIST используются без списка полей

Параметр	Значение параметра	Описание
	OFF	Пробелы вставляются
DELETE	ON	Записи, помещенные на удаление, не используются командами LIST, DISPLAY, COUNT
	OFF	Записи используются
HEADING TO	<строка>	<строка> длиной до 60 символов заполняется и печатается в качестве заголовка отчета
FORMAT TO	[SCREEN] [PRINT]	Определяет, куда идет выход команд @
	[<форм. файл>]	Определяет, что команды @ для команд READ считываются из <форм. файл>
DEFAULT TO	<дисковод>	Устанавливает для РЕБУСа, что все файлы, для которых не был явно указан дисковод, находятся на указанном в команде <дисковод>. Дисковод может быть задан в двух форматах — с двоеточием (как это принято в операционной системе SCP) или без него
ALTERNATE TO	<файл>	Позволяет записывать протокол работы пользователя с РЕБУСом в файл за исключением работы с командами, использующими полноэкранный режим. Эта команда должна предшествовать команде SET ALTERNATE ON
DATE TO	дд/мм/гг	Установка текущей даты
INDEX TO	<инд. файл1>], ..., <инд. файл7>]	Устанавливает от одного до семи индексных файлов SET INDEX TO без указания индексного файла приводит к тому, что все открытые ранее индексные файлы закрываются

режиме вам предоставляются большие возможности по вводу и корректировке данных. Так, при вводе данных в некоторую запись базы данных вы можете возвращаться к уже заполненным полям этой записи и корректировать введенную в них информацию, что нельзя сделать в обычном режиме.

Для каждой из команд СУБД в этом режиме существуют некоторые особенности использования команд полноэкранного редактирования. Перечень команд полноэкранного редактирования для команд РЕБУСа, работающих в этом режиме, приведены в табл. 6.14.

6.7. ПРОВЕРЯЕМ СЕБЯ. КАК УСВОЕН РЕБУС?

Упражнения

6.1. Каков минимальный состав файлов, необходимых для работы РЕБУСа?

6.2. Какой файл РЕБУСа требуется для получения справочной информации во время работы с этим пакетом программ?

6.3. Какая команда используется для получения справочной информации о РЕБУСе?

6.4. Какой командой надо воспользоваться для ввода описания структуры файла базы данных?

6.5. Каким образом к используемому файлу базы данных может быть добавлена «пустая» запись?

6.6. Сколько файлов базы данных могут одновременно обрабатываться РЕБУСом?

6.7. Сколько строк содержит таблица, полученная в результате выполнения операции прямого произведения двух таблиц?

6.8. Напишите последовательность команд РЕБУСа, выполняющих:

- соединение таблиц ПОДРАЗДЕЛЕНИЯ и СОТРУДНИКИ;
- проекцию полученной таблицы по столбцам: код под-

Таблица 6.14

Использование полноэкранных операций командами РЕБУСа

<table>
<tr><th rowspan="2">Полноэкранные операции</th><th colspan="7">Команды РЕБУСа</th></tr>
<tr><th>APPEND</th><th>INSERT</th><th>EDIT</th><th>CREATE</th><th>BROWSE</th><th>MODIFY</th><th>READ</th></tr>
<tr><td>Перемещение курсора в поле:</td><td colspan="7"></td></tr>
<tr><td>предыдущее</td><td colspan="7">∧E или ∧A</td></tr>
<tr><td>следующее</td><td colspan="7">∧X или ∧F</td></tr>
<tr><td>Перемещение курсора к символу:</td><td colspan="7"></td></tr>
<tr><td>предыдущему</td><td colspan="7">∧S</td></tr>
<tr><td>следующему</td><td colspan="7">∧D</td></tr>
<tr><td>Заполнение поля, в котором находится курсор</td><td colspan="7">∧Y</td></tr>
<tr><td>Переключение режимов вставка/замена</td><td colspan="7">∧V</td></tr>
<tr><td>Удалить символ:</td><td colspan="7"></td></tr>
<tr><td>в позиции курсора</td><td colspan="7">∧G</td></tr>
<tr><td>слева от курсора</td><td colspan="7"><DEL></td></tr>
<tr><td>Прекращение полноэкранного редактирования без сохранения сделанных изменений</td><td colspan="7">∧Q</td></tr>
</table>

Записать текущую запись и перейти к записи:							
следующей	∧C или ∧R	∧C или ∧R	∧C	∧C или ∧R	∧C	—	—
предыдущей	—	—	∧R	—	∧R	—	—
Установить/снять памятку удаления текущей записи	—	—	∧U	—	∧U	—	—
Записать текущую запись и начать редактирование	—	—	∧W	—	—	—	—
Записать данные на диск и закончить редактирование	—	—	—	—	∧W	∧W	—
Вставить пустую строку в позиции курсора	—	—	—	—	—	∧N	—
Удалить строку, в которой находится курсор	—	—	—	—	—	∧T	—
Переместить изображение на 1/4 экрана:							
вниз	—	—	—	—	—	∧C	—
вверх	—	—	—	—	—	∧R	—
Переместить изображение на одно поле:							
вправо	—	—	—	—	∧Z	—	—
влево	—	—	—	—	∧B	—	—

разделения; наименование подразделения; код сотрудника; фамилия, имя, отчество.

Результат поместите в файл базы данных с именем CHIEFS.

6.9. Какой командой РЕБУСа следует воспользоваться для выполнения командного файла EMPASK?

Ответы к упражнениям

6.1. Для работы с РЕБУСом необходимы как минимум два файла: REBUS.COM и REBUCOVR.COM.

6.2. Получение справочной информации во время работы с РЕБУСом возможно при наличии на диске файла REBUSMSG.TXT.

6.3. Вывод справочной информации о РЕБУСе производится по команде HELP.

6.4. Ввод описания структуры файла базы данных осуществляется по команде CREATE.

6.5. Добавление «пустой» записи к используемому файлу базы данных производится командой APPEND BLANK.

6.6. Одновременно РЕБУСом могут обрабатываться два файла базы данных в первичной и вторичной областях.

6.7. Количество строк в таблице, полученной в результате выполнения операции прямого произведения, определяется произведением количества строк в исходных таблицах.

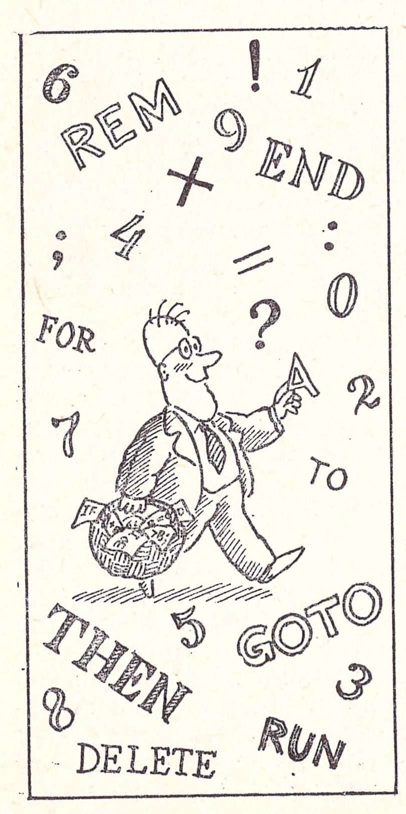
6.8. Последовательность команд, решающих задачу, такова:

```
SELECT SECONDARY
USE EMP
SELECT PRIMARY
USE DIV
JOIN TO CHIEFS FOR EMP:CODE=S.EMP:CODE;
FIELDS DIV:CODE, DIV:NAME, EMP:CODE;
S.EMP:NAME
```

6.9. Для выполнения командного файла EMPASK надо ввести команду DO EMPASK.

7

ЗНАКОМИМСЯ С ЯЗЫКОМ ПРОГРАММИРОВАНИЯ БЕЙСИК



7.1. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА

Что такое Бейсик?

Бейсик является языком программирования высокого уровня и служит средством записи программ, предназначенных для выполнения их с помощью компьютера. Он был создан в начале 60-х годов как язык обучения программированию. Легкость понимания этого языка достигается благодаря весьма простым правилам его построения и организации диалога между вами и компьютером в процессе создания программы. Бейсик, по сравнению с любым другим языком программирования, имеет большое число версий в зависимости от модели компьютера. Однако, зная одну из версий, вы легко перейдете на програм-

мирование в среду другой версии, так как основные конструкции являются неизменными.

В настоящей главе вы ознакомитесь с некоторыми конструкциями Бейсика для персонального компьютера Роботрон 1715 на примере решения несложных задач. Данная версия Бейсика может использовать как интерпретатор, так и компилятор. Компилятором советуем вам пользоваться только после приобретения навыка программирования на Бейсике. Осваивать язык, а также отлаживать программы целесообразно в режиме интерпретатора.

Как разрабатывается программа?

●●● Подготовка задачи к решению ее на компьютере начинается с определения целей решения и комплекта исходных данных. Далее выбирается математический метод решения и подготавливается математическое описание решаемой задачи. При этом устанавливается оптимальная применительно к данной задаче система обозначения всех переменных и параметров задачи.

●●● Теперь можно выполнить переход к формальному описанию информационного процесса, приводящего преобразования исходных данных к искомому результату. Такое формальное описание, выполненное по определенным и весьма несложным правилам, называется алгоритмом. Действия, предписанные алгоритмом, могут быть произведены человеком или компьютером. Понятно, что компьютер выполнит эти действия несравненно быстрее человека.

●●● Важным этапом работы при программировании является анализ данных, которые будут обрабатываться создаваемой программой. В разрабатываемой программе должны сообщаться некоторые сведения о свойствах используемых данных. Это позволит компьютеру правильно расположить данные в своей памяти, безошибочно извлекать их оттуда, выполнять только корректные операции с данными каждого типа.

●●● Наконец, можно приступить к кодированию про-

граммы, т. е. к записи разработанного алгоритма на выбранном языке программирования (в нашем случае — на Бейсике). Ваша работа по составлению программы, несомненно, сопровождалась ошибками. Это совершенно естественно.

Теперь важно устранить все ошибки в программе. Для этого понадобится осуществлять пробные решения составленной программы. Для поиска каждой отдельной ошибки может потребоваться специальный комплект исходных данных, применительно к которому результат работы программы вам точно известен. Отклонение результатов пробного решения от точного результата будет свидетельствовать о наличии в вашей программе одной или нескольких ошибок. Для установления характера и места возникновения ошибок вам потребуется проявить находчивость и определенную изобретательность. Совпадение результатов пробного пуска с точным результатом позволит надеяться, что программа, по-видимому, работает правильно.

Подобный процесс нахождения и устранения ошибок в программе, получивший название отладки программы, является весьма сложным, а зачастую и длительным. Успех здесь напрямую зависит от вашего опыта, аккуратности, внимательности, настойчивости и строгой дисциплины. В любом случае при возникновении трудностей с отладкой программы не следует отчаиваться, ведь вам «противостоит» исполнительный, аккуратный и очень быстродействующий формалист — компьютер, «поведение» которого абсолютно предсказуемо.

Алфавит Бейсика

Любой язык програмирования базируется на некоторой конечной совокупности символов — алфавите. В алфавит Бейсика входят:

- 26 прописных латинских букв от A до Z;
- 31 прописная русская буква от А до Я, исключая Ъ и Ё;

- десятичные цифры от 0 до 9;
- девять знаков операции + — * / \ = > < ∧;
- специальные символы . , : ; ! ? " ' ( ) % ' ¤ ! #.

Любая конструкция Бейсика состоит только из перечисленных символов. Использование какого-либо другого символа или невозможно, или приводит к так называемой синтаксической ошибке.

Синтаксис и семантика

Бейсик, как и все языки программирования, характеризуется синтаксисом и семантикой.

●●● Синтаксис задает способы построения правильных конструкций языка. На основании этих способов компьютер получает возможность отличать правильные конструкции от неправильных. При выполнении программы, имеющей нарушения синтаксических правил, компьютер фиксирует ошибку и, как правило, распознает ее.

●●● Семантика определяет характер и последовательность действий, предпринимаемых компьютером при выполнении каждого конкретного оператора программы.

У неопытного программиста чаще всего встречаются синтаксические ошибки. Устраняются они сравнительно просто, так как компьютер помогает в этом. По мере накопления опыта число синтаксических ошибок уменьшается. Устранение семантических ошибок, как правило, значительно сложнее и зависит от более глубокого знания и понимания работы каждой конструкции Бейсика.

Что такое программа на Бейсике?

Действия по обработке данных, которые выполняет компьютер, задаются с помощью последовательности операторов и команд Бейсика. Это основные конструкции языка, правила написания и использования которых вы должны хорошо понимать и помнить. С помощью операторов вы представляете алгоритм в форме, воспринимаемой компьютером. Команды позволяют вам управлять

работой компьютера в процессе создания, отладки и работы программы.

При записи операторов и команд используются специальные ключевые слова — английские слова или их сокращения, указывающие на выполняемые действия. Всего в Бейсике зарезервировано для этих целей 123 ключевых слова, например: FOR, IF, RUN, LIST и т. п.

Программа на языке Бейсик представляет собой последовательность операторов, записанных построчно. В каждой строке текста программы могут быть один или несколько операторов. Операторы строки отделяются друг от друга с помощью символа двоеточие. Строка программы не должна иметь более 255 символов и может занимать несколько строк экрана дисплея.

Каждая строка программы начинается с номера, который выражен целым числом без знака в диапазоне от 1 до 65529. В основной памяти компьютера строки программы располагаются по порядку своих номеров. Этот порядок не зависит от последовательности набора строк на экране дисплея при формировании текста программы. Удобно нумеровать строки программы с шагом 10, так как при добавлении в программу новых строк им присваиваются промежуточные номера, чтобы компьютер расположил их между существующими строками, не нарушая последовательности номеров.

Текст строки набирается при помощи клавиатуры с одновременным отображением его на экране дисплея. Ввод строки в память осуществляется после нажатия клавиши ЕТ. При этом соответствующий этой клавише символ на экране не отображается.

Какие данные обрабатываются программой на Бейсике?

Основная цель программы — обработка данных для определения значений исковых результатов. В качестве данных используются константы, переменные и массивы.

Компьютер выполняет операции над значениями, которые в программе могут быть представлены константами, простыми переменными (скалярами), переменными с индексами (элементы массивов) или указателями функций (например, указатели элементарных функций синуса, косинуса и т. п.).

Каждое данное и, следовательно, соответствующие ему значения могут быть отнесены либо к числовому (число), либо к символьному типу. Среди чисел в Бейсике различают целые, вещественные числа и числа удвоенной точности (другими словами, «длинные» числа). К символьному типу относятся данные, значения которых представляют собой цепочки (последовательности) символов алфавита Бейсика конечной длины.

Для обозначения того или иного типа используются специальные символы. Так, для обозначения переменной символьного типа после имени этой переменной записывается символ ¤, например: Н¤, Т5¤, ВН24¤. Символьная константа записывается в кавычках (апострофах) например: "БЕЙСИК", 'BASIC', "56.78".

Для обозначения числового типа переменных используются следующие символы:

% — целый тип, например: А%, N%, Р29%;

— тип удвоенной точности, например: HP#, LS#.

Если после имени переменной нет символа ¤, # или %, то такая переменная имеет тип вещественный.

Режимы работы компьютера с Бейсиком

Бейсиком предусматриваются два режима работы компьютера — непрограммный и программный.

●●● В непрограммном режиме используются только определенные операторы Бейсика. При этом номер строки должен отсутствовать. Бейсик выполняет каждый такой оператор немедленно после нажатия клавиши **ЕТ**. В этом режиме компьютер подобен калькулятору.

В программном режиме каждая строка программы име-

ет номер и после нажатия клавиши **ЕТ** производится только ее запись в память. Выполнения операторов введенной строки не происходит. Выполнение программы возможно только после записи ее в память и использования специальной команды запуска программы на решение — RUN.

7.2. РЕШАЕМ ЗАДАЧУ ОБРАБОТКИ ЧИСЛОВЫХ ДАННЫХ

7.2.1. СТАВИМ ЗАДАЧУ

Независимо от сложности задачи программа для ее решения с помощью компьютера состоит из трех частей:

- сообщение о назначении данной программы и основных ограничениях при использовании программы. Ввод исходных данных под управлением программы. Контроль вводимых значений и, как правило, вывод введенных значений исходных данных;
- выполнение собственно алгоритмической обработки исходных данных для получения искомого результата;
- оформление результатов обработки, исчерпывающее по содержанию и наиболее удобное по форме (текст, таблица, график, гистограмма и т. п.).

Для выработки правильной манеры программирования настоятельно рекомендуем всегда придерживаться этой схемы.

Пусть требуется решить следующую задачу: составить программу для нахождения площади параллелограмма, если известны длины его смежных сторон a и b и угол α между ними.

В этой задаче исходными данными являются, очевидно, три переменные, которые именуют собой три вещественных значения: длина стороны a, длина стороны b и угол α между сторонами. Результатом решения являет-

ся вещественное значение s площади геометрической фигуры, определенной, например, по формуле

$$s = ab \sin \alpha. \tag{7.1}$$

Выберем теперь имена (идентификаторы) для исходных данных и искомого результата, пользуясь алфавитом Бейсика. Пусть A — длина первой стороны, B — длина второй стороны, ALFA — угол между ними, S — искомая площадь.

Обратите внимание! В алфавите Бейсика отсутствует греческая буква альфа. В целях соблюдения математической традиции при обозначении углов и сохранения связи с математическим описанием задачи (7.1) целесообразно использовать сравнительно длинное имя ALFA.

Анализ свойств данных поставленной задачи позволяет сделать вывод о том, что все исходные данные и искомый результат относятся к вещественному типу.

Значение угла α может быть задано как в радианной, так и в градусной мере. Бейсик имеет ряд так называемых встроенных функций (синус, косинус, тангенс, логарифм и т. п.). При этом требуется, чтобы значение аргумента тригонометрических функций задавалось в радианной мере. По этой причине исходное значение для переменной ALFA нашей задачи должно задаваться в радианах, хотя, как будет показано далее, оно может быть задано и в угловой мере.

7.2.2. СОСТАВЛЯЕМ ПРОСТЕЙШИЕ ПРОГРАММЫ

Первый вариант — используем простейшие операторы

Приступим теперь к записи операторов Бейсика, которые будут определять действия компьютера. Сначала

исходные данные должны быть введены в основную память компьютера. Это достигается применением оператора ввода INPUT, в котором указывается список вводимых переменных:

10 INPUT A, B, ALFA

Вводимое с клавиатуры первое числовое значение присваивается переменной A, второе значение — переменной B и третье значение — переменной ALFA. Вводимые значения вы можете видеть на экране дисплея.

Применяя основной вычислительный оператор Бейсика — оператор присваивания, вы можете предписать компьютеру выполнение необходимых вычислений в соответствии с выражением (7.1). Внешне этот оператор напоминает математическое выражение

20S = A * B*SIN (ALFA)

Обратите внимание! Знак операции умножения изображается символом звездочка, а аргумент функции всегда заключается в скобки.

Наконец, необходимо вывести вычисленное значение площади из памяти на экран дисплея. Это выполняется при помощи оператора вывода PRINT и списка выводимых переменных. В данном случае выводится только одно значение

30 PRINT S

Запишите теперь все эти операторы в виде Бейсик-программы. При этом следует использовать еще один оператор END, которым полагается заканчивать любую программу. Перед вами ваша первая программа. Попробуйте теперь выполнить ее на вашем компьютере.

●●● Убедитесь, что компьютер и печатающее устройство (принтер) включены. Правильно вставьте диск в левый дисковод и закройте замок. Этих действий достаточно, чтобы хранящаяся на дискете операционная система была загружена (переписана с диска) в основную память компьютера. При благоприятном исходе операции загрузки вы увидите на экране первоначальное сообщение

операционной системы, которое в последней своей строке имеет два символа:

A>

●●● Можно приступать к работе с Бейсиком. Сначала необходимо загрузить интерпретатор Бейсика. Для этого с помощью клавиатуры наберите строку BASIC и нажмите клавишу **ET**. Окончание загрузки интерпретатора Бейсика сопровождается выдачей на экран нескольких строк сообщения. В последней строке этого сообщения вы увидите два символа:

OK

Таким образом Бейсик сообщает о своей готовности работать с вами.

●●● Произведите набор четырех строк вашей программы, заканчивая набор каждой строки нажатием клавиши **ET**. Для запуска программы воспользуйтесь командой RUN.

●●● Программа запущена. В начале следующей строки экрана вы увидите вопросительный знак. Это оператор INPUT ожидает от вас ввода трех числовых значений. Наберите, например, значения 2, 4 и 0.25 в одну строку через запятую (не забывайте заканчивать набор строки нажатием клавиши **ET**).

●●● Вы увидите на экране число 1.97923, которое и является искомой площадью параллелограмма. Выводом символов OK Бейсик сообщает вам о готовности продолжать работу.

Посмотрите на стр. 325, как будет выглядеть экран, отражающий все ваши действия с компьютером.

Обратите внимание! Здесь и в дальнейших примерах вводимые вами команды и значения подчеркнуты.

Вашу первую программу вполне можно использовать для вычисления площади параллелограмма. Однако на практике вы вскоре убедитесь, что пользоваться такой программой очень неудобно. Например, вы можете забыть,

Пример 7.1.

```
A>BASIC
BASIC-80 REV. 5.2
[CP/M VERSION]
COPYRIGHT 1977, 78, 79, 80 (C) BY MICROSOFT
CREATED: 14-JUL-80
24390 BYTES FREE
OK
10 INPUT A,B,ALFA
20 S=A*B*SIN(ALFA)
30 PRINT S
40 END
RUN
? 2,4,0.25
 1.97923
OK
```

в каком порядке следует набирать вводимые значения, искомый результат нуждается в выделении среди текста на экране дисплея.

Второй вариант — используем поясняющий текст

Попробуем улучшить вашу программу введением в нее символьных констант. Эти константы будут играть роль поясняющего текста, который выводится на экран во время выполнения программы. Символьные константы можно использовать как в операторе ввода, так и в операторе вывода.

Теперь экран с вторым вариантом программы и результатом ее работы будет выглядеть как на стр. 326.

Согласитесь, с такой программой работать значительно приятнее. Она сама помогает вам избежать ошибок при вводе значений, а результат вычислений достаточно хорошо выделен. Такая программа во время своей работы подсказывает вам, ввод какого значения она ожидает от вас.

П р и м е р 7.2.

```
OK
10 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
20 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
30 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ",ALFA
40 S=A*B*SIN(ALFA)
50 PRINT"ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА";S
60 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ .25
ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА 1.97923
OK
```

7.2.3. ИСПОЛЬЗУЕМ В ПРОГРАММЕ ОПЕРАТОРЫ ПЕРЕХОДА

Какие недостатки имеет ваша программа?

Любую программу можно совершенствовать практически до бесконечности. Ваша программа, признаемся, еще очень далека от идеала. Какие недостатки можно обнаружить в программе?

●●● Во-первых, в начале программы весьма полезно иметь информационный текст, сообщающий основное назначение данной программы. Без него желание знать, для чего предназначена программа, заставит вас внимательно просматривать весь текст программы и вникать в его смысл. Полезно также иметь информацию о всех или хотя бы основных ограничениях при решении данной задачи.

●●● Во-вторых, ваша программа не защищена от ввода «глупых» (некорректных) значений исходных данных. В самом деле, длина отрезка никогда не может быть от-

рицательной. Значение угла α тоже должно быть неотрицательным числом, иначе даже при положительных значениях длин сторон a и b площадь s может получиться отрицательной.

Первый из двух отмеченных недостатков может быть устранен при помощи оператора REM, который позволяет вносить в текст программы любые комментарии, пояснения. Оператор может занимать любое место в программе и никак не сказывается на проводимых по программе вычислениях.

Контроль вводимых значений можете выполнить при помощи условного оператора IF, который позволяет изменять направление вычислительного процесса в зависимости от факта выполнения или невыполнения условия, содержащегося в этом операторе. Здесь же может понадобиться и оператор GOTO безусловного перехода. Он нарушает естественную последовательность выполнения операторов программы в любом случае (безусловно).

Как записать условный оператор?

Условный оператор записывается в виде (формате)

IF условие THEN операторы [ELSE операторы]

Здесь и далее при описании синтаксической структуры конструкций Бейсика будут использоваться такие символы, не входящие в его алфавит:

[ ] — часть конструкции, заключенная в квадратные скобки, является необязательной и может быть отброшена;

{ } — из указанных в фигурных скобках фрагментов конструкции может использоваться любой.

Оператор IF должен размещаться в одной строке программы. Выполняется он следующим образом. Анализируется условие, которое может быть истинным или ложным. При соблюдении условия выполняются операторы,

следующие за ключевым словом THEN, а затем — операторы следующей строки. При несоблюдении условия будут выполняться операторы, следующие за ключевым словом ELSE, и затем — операторы следующей строки.

Часто используется сокращенная форма оператора IF, когда в нем отсутствует часть ELSE с соответствующей группой операторов. Бейсик допускает также использование вложенных условных операторов.

Как записать оператор безусловного перехода?

Оператор безусловного перехода записывается в виде

GOTO номер строки

Этот оператор явно определяет своего преемника: после оператора GOTO будет выполняться первый оператор строки, номер которой указан в операторе безусловного перехода.

Составляем программу с операторами перехода

Вернемся теперь к вашей программе. Попробуйте записать операторы Бейсика, которые вводят значение длины одной из сторон и проверяют его знак. Если введенное значение оказывается отрицательным, то на экране должны появиться сообщение "Вы ошиблись" и вновь запрос "Введите длину первой стороны" на ввод значения длины этой же стороны.

Проверьте себя. Вот как должны выглядеть необходимые операторы:

```
10 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
20 IF A<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 10
```

Еще раз проанализируйте работу этих операторов. Оператор ввода в строке 10 выводит на экран текст подсказки "Введите длину первой стороны" и ожидает от вас набора на клавиатуре определенного числового значения. После окончания набора вводимого значения и нажатия клавиши **ET** оно поступает в память компьютера. Начинает работу оператор IF в строке 20. Он извлекает из памяти только что введенное значение переменной A и анализирует его знак (A<0). Если окажется, что A действительно меньше нуля, то будут выполнены операторы, записанные после ключевого слова THEN. Сначала на экране появится сообщение "Вы ошиблись", а затем управление с помощью оператора GOTO будет передано оператору ввода в строке 10. И вновь на экране появится сообщение "Введите длину первой стороны".

Если при анализе условный оператор в строке 20 об-

Пример 7.3.

```
OK
10 REM ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА
20 REM ПО ДВУМ СТОРОНАМ И УГЛУ МЕЖДУ НИМИ
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 IF A<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 30
50 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
60 IF B<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 50
70 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ",ALFA
80 IF ALFA<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 70
90 S=A*B*SIN(ALFA)
100 PRINT"ПЛОЩАДЬ ПАРАЛЛЕЛОГАММА РАВНА";S
110 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ -4
ВЫ ОШИБЛИСЬ
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ .25
ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА 1.97923
OK
```

наружит, что значение A не меньше нуля, то операторы PRINT и GOTO выполняться не будут.

Попробуйте получить на экране полный текст усовершенствованной программы с комментариями и проверкой вводимых значений. Сравните свою работу с примером 7.3.

В примере намеренно сделана попытка ввести недопустимое отрицательное значение минус 4 для переменной B. Реакция программы соответствует нашему замыслу.

7.2.4. ИСПОЛЬЗУЕМ СИМВОЛЬНЫЕ ПЕРЕМЕННЫЕ

Как сделать, чтобы угол можно было вводить в любой мере?

На следующем этапе хотелось бы сделать программу универсальной в том смысле, что она будет способна вычислять правильно площадь параллелограмма при любой — градусной или радианной — мере угла α. Если ранее вы ознакомились с применением так называемых подсказок, которые компьютер выдает вам на экран во время выполнения программы, то сейчас вы узнаете как организуется диалог «компьютер — пользователь». При этом вы ознакомитесь с одним из способов использования символьных переменных.

Вспомните! Перевод значения $\alpha_г$ в градусной мере в эквивалентное значение $\alpha_р$, выраженное в радианах, выполняется по формуле

$$\alpha_р = \frac{\pi}{180}\alpha_г = \frac{3{,}1415}{180}\alpha_г. \qquad (7.2)$$

Сделаем так, чтобы при работающей программе после ввода значения для переменной ALFA компьютер спросил вас (выводом на экран текста) «Угол измерен в градусах?». При этом требуется указание правильных вариантов ответа пользователя «Да/Нет». В ответ на такой запрос вы должны ввести с клавиатуры либо ДА, либо НЕТ.

В первом случае, очевидно, нужно воспользоваться формулой (7.2) перевода, а во втором случае ничего предпринимать не следует. На любой другой ответ пользователя компьютер должен реагировать выводом сообщения «Я вас не понял» и повторением вопроса «Угол измерен в градусах?».

Для осуществления наших планов воспользуйтесь в программе вспомогательной переменной символьного типа, например H¤. Эта переменная может принимать только символьные значения, к которым относятся ДА и НЕТ.

Очередная версия вашей программы и результат ее работы могут выглядеть так, как показано ниже.

Пример 7.4.

```
OK
10 REM ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА
20 REM ПО ДВУМ СТОРОНАМ И УГЛУ МЕЖДУ НИМИ
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 IF A<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 30
50 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
60 IF B<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 50
70 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА",ALFA
80 IF ALFA<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 70
90 INPUT"УГОЛ ИЗМЕРЕН В ГРАДУСАХ? (ДА/НЕТ)",H¤
100 IF H¤="ДА" THEN ALFA=ALFA*3.1415/180:GOTO 120
110 IF H¤<>"НЕТ" THEN PRINT"Я ВАС НЕ ПОНЯЛ":GOTO 90
120 S=A*B*SIN(ALFA)
130 PRINT"ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА",S
140 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ -2
ВЫ ОШИБЛИСЬ
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА .25
УГОЛ ИЗМЕРЕН В ГРАДУСАХ? (ДА/НЕТ) NO
Я ВАС НЕ ПОНЯЛ
УГОЛ ИЗМЕРЕН В ГРАДУСАХ? (ДА/НЕТ) НЕТ
ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА 1.97923
OK
```

В этой программе новыми являются строки 90—110. Оператор ввода в строке 90 выводит на экран текст вопроса «Угол измерен в градусах?» и правильные варианты вашего ответа «(Да/Нет)». Ответ пользователя, представляющий собой символьное значение, присваивается переменной H¤, указанной в операторе INPUT строки 90.

●●● Если пользователь на вопрос отвечает набором на клавиатуре значения ДА, то условие в операторе IF строки 100 становится истинным. Выполняются оператор присваивания, размещенный после ключевого слова THEN, и оператор GOTO. Последний передает управление оператору присваивания строки 120.

Обратите внимание! Использование одной и той же переменной ALFA слева и справа от символа = является совершенно правильным. Оператор присваивания выражает временной характер выполняемых вычислений: сначала значение переменной ALFA извлекается из памяти, затем оно умножается на константу 3.1415, полученный результат делится на константу 180 и полученное значение присваивается переменной ALFA, стоящей слева от знака =. Таким образом, в результате выполнения оператора присваивания значение переменной ALFA изменилось.

●●● Если пользователь на вопрос «Угол измерен в градусах?» отвечает набором на клавиатуре значения НЕТ, то условие в операторе IF строки 100 становится ложным. Значение переменной ALFA в этом случае не изменяется, и начинает выполняться оператор IF строки 110.

Обратите внимание! Операция деления обозначается с помощью символа /, а знак < > означает «не равно». При записи вещественной константы 3.1415 для отделения целой части от дробной используется символ точка, а не запятая.

Переменная H¤, по-прежнему имеет значение НЕТ. Следовательно, условие в операторе IF строки 110 является ложным и остальная часть условного оператора не выполняется. Далее будет выполняться оператор присваивания строки 120.

●●● Если пользователь на вопрос «Угол измерен в градусах?» отвечает непредусмотренным значением, например NO, то условие в операторе IF строки 100 становится лож-

ным и далее начинает выполняться оператор IF строки 110. В этом операторе условие приобретает истинное значение, так как значение NO переменной Н ¤ не равно значению НЕТ. Следовательно, будет выполнен оператор вывода текста «Я вас не понял» и следующим будет выполняться оператор ввода строки 90 из-за действия оператора GOTO 90.

7.2.5. ОРГАНИЗУЕМ ДОКУМЕНТИРОВАНИЕ РАБОТЫ ПРОГРАММЫ

Работаем с принтером

Для оформления результатов работы программы в виде документа необходимо обеспечить вывод информации на принтер. Для этого используется оператор вывода LPRINT, аналогичный оператору PRINT.

Направив результаты вычислений не на экран, а на принтер, вы должны заметить, что возникло определенное затруднение: значения исходных данных, для которых производится расчет, отображаются на экране, а результаты вычислений — на бумажном листе. Информация о значениях исходных данных, имеющаяся на экране, в процессе вашей работы будет утеряна, так как содержимое экрана постоянно сдвигается вверх. Результаты работы программы на бумаге без соответствующих значений исходных данных не представляют практической ценности. Возникает необходимость несколько изменить вывод данных в вашей программе для получения исчерпывающей информации о работе программы.

Важно подчеркнуть, что все подсказки и диалоговый ввод данных реализуются по-прежнему с помощью экрана дисплея. Наоборот, в результате работы программы должен быть получен бумажный документ (распечатка), содержащий исчерпывающую информацию о проведенном расчете, в том числе об исходных данных и результатах.

Для достижения поставленной цели к уже существующему оператору вывода искомой площади добавьте операторы вывода значений исходных данных, для которых и получено решение. Полезно также несколько изменить форму вопроса в операторе ввода строки 90 и оба варианта ответа для обеспечения вывода наименования единицы угла α. Разумеется несколько изменятся условия в операторах IF строк 100 и 110.

После модернизации ваша программа и результат ее работы могут быть такими.

Пример 7.5.

```
OK
10 REM ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА
20 REM ПО ДВУМ СТОРОНАМ И УГЛУ МЕЖДУ НИМИ
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 IF A<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 30
50 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
60 IF B<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 50
70 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА",ALFA
80 IF ALFA<0 THEN PRINT"ВЫ ОШИБЛИСЬ":GOTO 70
90 INPUT"В КАКИХ ЕДИНИЦАХ ИЗМЕРЕН УГОЛ? (ГРАД/РАД)",H¤
100 IF H¤="ГРАД" THEN ALFA=ALFA*3.1415/180:GOTO 120
110 IF H¤<>"РАД" THEN PRINT"Я ВАС НЕ ПОНЯЛ":GOTO 90
120 S=A*B*SIN(ALFA)
130 LPRINT
140 LPRINT"ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА";S
150 LPRINT"ПРИ ДЛИНАХ СТОРОН";A;"И";B
160 LPRINT"И УГЛУ МЕЖДУ НИМИ";ALFA;H¤
170 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА .25
В КАКИХ ЕДИНИЦАХ ИЗМЕРЕН УГОЛ? (ГРАД/РАД) RAD
Я ВАС НЕ ПОНЯЛ
В КАКИХ ЕДИНИЦАХ ИЗМЕРЕН УГОЛ? (ГРАД/РАД) РАД
OK


ПЛОЩАДЬ ПАРАЛЛЕЛОГРАММА РАВНА 1.97923
ПРИ ДЛИНАХ СТОРОН 2 И 4
И УГЛУ МЕЖДУ НИМИ .25 РАД
```

Для лучшего отделения при выводе результата решения задачи полезно использовать оператор вывода без списка вывода (строка 130). В этом случае при выводе пропускается одна строка.

Обратите внимание! Имеются два способа разделения элементов списка вывода в операторе вывода. При использовании запятой осуществляется зонный вывод, когда каждое выводимое значение размещается в своей зоне. Под каждую зону отводятся 11 позиций, между зонами оставляются три пробела. При использовании точки с запятой или пробела происходит последовательный вывод, при котором очередное выводимое значение отделяется от предыдущего одним пробелом. При превышении формата строки вывода (80 позиций) автоматически выполняется переход на новую строку и вывод продолжается.

7.2.6. РАЗРАБАТЫВАЕМ ЦИКЛИЧЕСКУЮ ПРОГРАММУ

Зачем нужна циклическая программа?

Разработанная вами программа выполняет однократный расчет площади геометрической фигуры. Для повторного расчета вам придется вновь запускать программу командой RUN. При этом ее текст опять обрабатывается интерпретатором, что приводит к непроизводительной трате времени. Следовательно, если имеется потребность выполнять серию вычислений площади параллелограмма, вы должны видоизменить свою программу, придав ей циклический характер.

Циклом называется фрагмент алгоритма или программы, повторяющийся более одного раза. Различают циклы с заранее известным числом повторений (детерминированные) и циклы, число повторений которых заранее не известно (итерационные). Для организации циклического

процесса необходимо обеспечить требуемое число его повторений и выделение группы операторов, выполняющихся многократно и называемых телом цикла.

Организуем цикл с помощью операторов перехода

Сформировать циклический вычислительный процесс можно при помощи уже известных вам операторов перехода — оператора IF и оператора GOTO.

Таблица 7.1

Зонный формат для оформления результатов вычислений

11	3	11	3	11	3	11	3	11
Номер		Длина А		Длина В		Угол, рад		Площадь

Пусть требуется проводить неоднократные вычисления площади параллелограмма, причем заранее не известно количество таких вычислений. Стало быть, речь идет об организации итерационного вычислительного процесса. Поскольку вычисления площади будут проводиться многократно, то и результатов будет целая серия. Отсюда делаем заключение о необходимости оформления результатов вычислений в табличном виде, используя зонный формат (табл. 7.1). Заметим, что каждое выводимое в зону значение прижимается к ее левой границе.

Для того чтобы разрабатываемая программа не была бы слишком велика, не будем в этот раз выполнять про-

верку вводимых значений и потребуем, чтобы значение угла α вводилось бы только в радианной мере. В дальнейшем, мы надеемся, вам не составит большого труда учесть в циклической программе опущенные подробности.

Приступая к разработке циклической программы, полезно спланировать последовательность програмирования отдельных этапов всего вычислительного процесса. Удобно сделать это в виде схемы алгоритма (рис. 7.1). Для подсчета количества вычисленных значений площади используйте, например, переменную N целого типа. До начала циклических вычислений ее значение должно быть равно нулю. Для хранения ответа пользователя «ДА» или «НЕТ» на вопрос компьютера о необходимости продолжения вычислений используйте, например, переменную T.

Посмотрите на стр. 338, как может выглядеть ваша циклическая программа с примером экранного диалога и результатами решения, выводимыми на принтер в табличном виде.

В программе используются знакомые вам операторы. Цикл повторяется три раза для трех комплектов исходных данных.

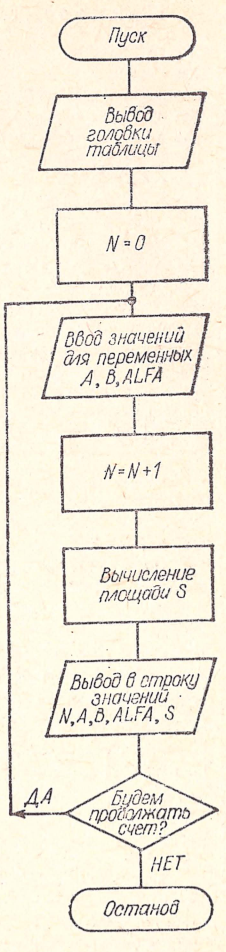


Рис. 7.1. Схема алгоритма итерационного цикла

Организуем цикл с помощью оператора **WHILE**

В Бейсике имеется специальная конструкция для описания итерационных

Пример 7.6.

```
OK
10 REM ЦИКЛИЧЕСКАЯ ИТЕРАЦИОННАЯ ПРОГРАММА
20 LPRINT"-----------------------------------------------------------------------"
30 LPRINT"НОМЕР","ДЛИНА A","ДЛИНА B","УГОЛ,РАД ","ПЛОЩАДЬ"
40 LPRINT"-----------------------------------------------------------------------"
50 N%=0
60 REM НАЧАЛО ЦИКЛА
70 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
80 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
90 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ",ALFA
100 N%=N%+1
110 S=A*B*SIN(ALFA)
120 LPRINT N%,A,B,ALFA,S
130 INPUT"БУДЕМ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ)",T$
140 IF T$="ДА" THEN GOTO 70
150 IF T$<>"НЕТ" THEN PRINT"ВАШ ОТВЕТ НЕПОНЯТЕН!":GOTO 130
160 LPRINT"----------------------------------------------------------------------"
170 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ .25
БУДЕМ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ) ДА
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2.2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4.2
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ .25
БУДЕМ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ) ДА
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2.6
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4.6
ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ .3
БУДЕМ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ) НЕТ
OK
-----------------------------------------------------------------------
НОМЕР    ДЛИНА A          ДЛИНА B          УГОЛ,РАД        ПЛОЩАДЬ
-----------------------------------------------------------------------
 1        2                4                .25             1.97923
 2        2.2              4.2              .25             2.28601
 3        2.6              4.6              .3              3.53442
-----------------------------------------------------------------------
```

вычислительных процессов — оператор WHILE. Он имеет вид

```
WHILE условие
{Операторы тела цикла
WEND
```

В этом операторе условие является отношением или числовым выражением. Работа оператора протекает следующим образом. Сначала проверяется значение условия. Если оно истинно в случае отношения (или не равно нулю в случае числового выражения), то выполняются операторы тела цикла. Среди последних должны быть операторы, изменяющие значения переменных условия. Вновь проверяется условие. Если оно оказывается ложным (или нулевым), то выполнение цикла заканчивается и управление передается оператору, следующему за ключевым словом WEND.

Таким образом, оператор WHILE обеспечивает повторение тела цикла нуль или более раз.

Ниже приводится текст программы с использованием оператора WHILE. Циклические вычисления площади продолжаются до тех пор, пока пользователь на запрос компьютера «Будем продолжать вычисления?» будет отвечать «ДА». Любой другой ответ приводит к окончанию работы программы. Результаты решения, получаемые на принтере, оформлены так же, как и в предыдущем примере.

Присваивание переменной Т¤ значения ДА в строке 70 требуется для обеспечения начала работы оператора WHILE.

Обратите внимание! В операторе присваивания строки 70 символ — выражает факт присваивания значения ДА переменной Т¤, а в условии оператора WHILE строки 90 этот же символ является знаком операции отношения «равно».

Пример 7.7.

```
OK
10 REM ЦИКЛИЧЕСКАЯ ИТЕРАЦИОННАЯ ПРОГРАММА
20 REM С ОПЕРАТОРОМ ЦИКЛА WHILE
30 LPRINT"-----------------------------------------------------------------"
40 LPRINT"НОМЕР","ДЛИНА A","ДЛИНА B","УГОЛ,РАД ","ПЛОЩАДЬ"
50 LPRINT"-----------------------------------------------------------------"
60 N%=0
70 T$="ДА"
80 REM НАЧАЛО ЦИКЛА
90 WHILE T$="ДА"
100 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
110 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
120 INPUT"ВВЕДИТЕ ЗНАЧЕНИЕ УГЛА В РАДИАНАХ",ALFA
130 N%=N%+1
140 S=A*B*SIN(ALFA)
150 LPRINT N%,A,B,ALFA,S
160 INPUT"БУДЕМ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ)",T$
170 WEND
180 LPRINT"---------------------------------------------------------------"
190 END
RUN
```

Организуем цикл с помощью оператора FOR

Вновь несколько видоизменим условие нашей задачи. Пусть требуется вычислять площадь параллелограмма при условии, что длины обеих сторон не изменяются, а угол α изменяется от α_0 до $\alpha_к$ с шагом $\Delta\alpha\,(\alpha_к > \alpha_0)$. В этом случае программа остается циклической, но теперь заранее известно необходимое число повторений:

$$N = \left]\frac{\alpha_к - \alpha_0}{\Delta\alpha}\right[+ 1, \tag{7.3}$$

где]...[— операция отбрасывания дробной части числа.

Для программирования подобных задач целесообразно воспользоваться оператором FOR организации детерминированных циклов. Этот оператор имеет вид

```
FOR пар. цикла = нач. зн. TO кон. зн. [STEP шаг]
{Операторы тела цикла
NEXT [пар. цикла]
```

Здесь пар.цикла — простая переменная, называемая параметром цикла и имеющая целый или вещественный тип; нач.зн., кон.зн., шаг — начальное, конечное значения и шаг изменения параметра цикла, каждое из которых задается числовым выражением целого или вещественного типа. Если шаг изменения параметра цикла не задан, то он принимается равным единице.

Оператор FOR обеспечивает следующий порядок работы. Сначала вычисляются начальное, конечное значения и шаг изменения параметра цикла. В дальнейшем в течение всего периода выполнения оператора цикла эти три значения не могут быть изменены. Следовательно, переход к операторам тела цикла, минуя заголовок цикла FOR-TO-STEP, недопустим.

Параметру цикла присваивается вычисленное начальное значение. Выполняются операторы тела цикла. Значение параметра цикла изменяется на величину, равную значению шага изменения. Следующий проход цикла происходит в том случае, если новое значение параметра цикла не превысит (при положительном шаге) или не станет меньше (при отрицательном шаге) конечного значения. Таким образом, оператор FOR заставляет выполняться операторы тела цикла минимум один раз.

Целесообразно для данного случая изменить способ оформления ре-

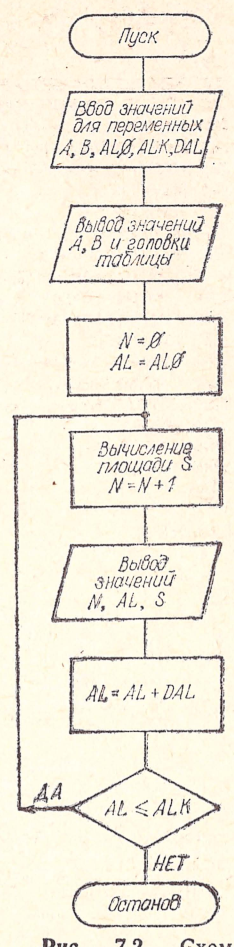


Рис. 7.2. Схема алгоритма детерминированного цикла

Пример 7.8.

```
OK
10 REM ЦИКЛИЧЕСКАЯ ПРОГРАММА
20 REM С ОПЕРАТОРОМ ЦИКЛА FOR
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
50 PRINT"ВВЕДИТЕ ДЛЯ УГЛА АЛЬФА В РАДИАНАХ:"
60 INPUT"      -НАЧАЛЬНОЕ ЗНАЧЕНИЕ",ALO
70 INPUT"      -КОНЕЧНОЕ  ЗНАЧЕНИЕ",ALK
80 INPUT"      -ШАГ ИЗМЕНЕНИЯ",DAL
90 REM ПОДГОТОВКА "ГОЛОВКИ" ТАБЛИЦЫ
100 LPRINT"ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА"
110 LPRINT"ПРИ ДЛИНАХ СТОРОН";A;"И";B
120 LPRINT"УГОЛ МЕНЯЕТСЯ ОТ";ALO;"ДО";ALK
130 LPRINT"С ШАГОМ";DAL
140 LPRINT"------------------------------------------------"
150 LPRINT"НОМЕР","УГОЛ,РАД ","ПЛОЩАДЬ"
160 LPRINT"------------------------------------------------"
170 N%=0
180 W=A*B
190 FOR ALFA=ALO TO ALK STEP DAL
200 N%=N%+1
210 LPRINT N%,ALFA,W*SIN(ALFA)
220 NEXT
230 LPRINT"------------------------------------------------"
240 END
RUN
ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ 2
ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ 4
ВВЕДИТЕ ДЛЯ УГЛА АЛЬФА В РАДИАНАХ:
      -НАЧАЛЬНОЕ ЗНАЧЕНИЕ 0
      -КОНЕЧНОЕ ЗНАЧЕНИЕ 1
      -ШАГ ИЗМЕНЕНИЯ .25
OK
ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА
ПРИ ДЛИНАХ СТОРОН 2 И 4
УГОЛ МЕНЯЕТСЯ ОТ 0 ДО 1
С ШАГОМ .25
------------------------------------------------
НОМЕР          УГОЛ,РАД       ПЛОЩАДЬ
------------------------------------------------
 1              0              0
 2              .25            1.97923
 3              .5             3.8354
 4              .75            5.45311
 5              1              6.73177
------------------------------------------------
```

зультатов вычислений, так как значения длин сторон меняться не будут. Можно, например, вынести длины сторон в заголовок таблицы, здесь же поместить сведения о диапазоне изменения угла α, а таблицу сформировать из трех граф: «Номер», «Угол, рад» и «Площадь».

Полезно разработать сначала схему алгоритма (рис. 7.2). Для новых переменных задачи можно, например, принять такие имена: α_0 назвать ALØ, $\alpha_к$—ALK и $\Delta\alpha$—DAL.

Ваша новая программа может иметь вид, показанный на стр. 342. Приводятся вариант экранного диалога и результат работы программы, получаемый на принтере.

7.2.7 КАК ИСПОЛЬЗУЮТСЯ ФУНКЦИИ И ПОДПРОГРАММЫ?

Воспользуйтесь функцией

Разрабатывая свою программу, вы можете не только использовать встроенные функции, имеющиеся в Бейсике, но и конструировать свои функции в целях получения эффективной программы. Встроенные функции — это программы для вычисления значений математических функций, хранящиеся в библиотеке Бейсика, например: программы вычисления экспоненты, синуса, косинуса. Список встроенных функций приведен в табл. 7.2.

Ввести в программу свою функцию вы можете с помощью оператора DEF описания функции, имеющего вид

DEF FN имя (список формальных параметров) = выражение

Этот оператор должен всегда располагаться в программе перед первым оператором, использующим соответствую-

Таблица 7.2

Встроенные числовые функции Бейсика

Имя	Назначение	Математическая запись	Тип функции
ABS	Абсолютное значение	$y = \|x\|$	Целый, вещественный
ATN	Арктангенс	$y = \mathrm{arctg}\ (x),\ -\pi/2 \leqslant y \leqslant \pi/2$	Вещественный
COS	Косинус аргумента в радианной мере	$y = \cos\ (x)$	
EXP	Экспоненциальная функция	$y = e^x,\ x \leqslant 87{,}3365$	
FIX	Целая часть значения, полученная отбрасыванием дробной части	—	Целый
INT	Целая часть значения, полученная округлением	—	
LOG	Натуральный логарифм	$y = \ln\ (x),\ x > 0$	Вещественный
RND	Псевдослучайное число в диапазоне от 0 до 1	—	
SGN	Знак аргумента	$y(x) = \begin{cases} +1 & \text{при } x > 0 \\ 0 & \text{„ } x = 0 \\ -1 & \text{„ } x < 0 \end{cases}$	Целый
SIN	Синус аргумента в радианной мере	$y = \sin\ (x)$	Вещественный
SQR	Квадратный корень	$y = \sqrt{x},\ x \geqslant 0$	
TAN	Тангенс аргумента в радианной мере	$y = \mathrm{tg}\ (x), -\pi/2 < x < \pi/2$	

Примечание. Аргумент каждой функции может принимать значения любых числовых типов (вещественного, целого, удвоенной точности).

щий указатель функции. Две буквы FN образуют постоянную часть имени функции, определяемой пользователем.

Важно подчеркнуть, что оператор DEF показывает, как вычислить единственное значение определяемой пользователем функции. Реализация этих вычислений осуществляется, как обычно, с помощью соответствующего указателя функции. Тип значения функции, определяемой оператором DEF, задается обычным способом. Длина оператора DEF не должна превышать 255 символов.

Пусть, например, требуется вычислить значение функции

$$P = \frac{A}{1 + \frac{B}{1 + C}}.$$

Это можно сделать, используя оператор DEF:

```
10 DEF FNF(X,Y)=X/(1+Y)
20 INPUT A,B,C
30 P=FNF(A,FNF(B,C))
40 PRINT P
50 END
```

При исходных значениях A=10, B=16 и C=3 значение функции P будет равно двум. Переменные X и Y в операторе DEF называются формальными параметрами, а переменные B, C, A и указатель функции FNF(B, C) в строке 30 — фактическими параметрами.

Использование в программе переменных, являющихся формальными параметрами в операторе DEF, не приводит к ошибке.

Воспользуйтесь подпрограммой

Бейсик, как и любой другой язык программирования высокого уровня, имеет способ описания многократно ис-

пользуемой в разных местах программы совокупности действий — подпрограммы. Подпрограммы играют подчиненную роль по отношению к программе, в которой они используются.

Процесс использования подпрограммы распадается на две части: описание подпрограммы и вызов подпрограммы. Описание подпрограммы (или просто подпрограмма) представляет собой обычный текст на Бейсике, причем номера строк подпрограммы соответствуют местоположению этой подпрограммы в тексте основной программы. Начальная строка подпрограммы не имеет каких-либо особенностей, отличающих ее от других строк основной программы. Заканчивается подпрограмма специальным оператором RETURN. В одной подпрограмме может встречаться несколько операторов RETURN. Таким образом, подпрограмма в отличие от оператора DEF не имеет механизма передачи фактических параметров и замены ими формальных параметров в описательной части конструкции.

Вызов подпрограммы из основной программы осуществляется с помощью оператора GOSUB, имеющего вид

GOSUB номер строки

После ключевого слова GOSUB указывается номер строки, с которой должна стартовать подпрограмма. Существенно, что каждая отдельная подпрограмма может иметь несколько стартовых строк (точек входа).

Оператор RETURN в конце подпрограммы действует как оператор безусловного перехода, возвращая управление первому оператору строки, непосредственно следующей за оператором GOSUB.

Составьте программу, используя функцию и подпрограмму

Попробуйте использовать функцию и подпрограмму в своей программе вычисления площади параллелограмма.

С помощью функции можно вычислить площадь фигуры, а подпрограмму используйте для проведения горизонтальных линий при выводе таблицы.

Ваша новая программа может выглядеть так.

Пример 7.9.

```
10 REM ИСПОЛЬЗОВАНИЕ ФУНКЦИИ И ПОДПРОГРАММЫ
20 DEF FNS(X)=A*B*SIN(X)
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
50 PRINT"ВВЕДИТЕ ДЛЯ УГЛА АЛЬФА В РАДИАНАХ:"
60 INPUT"       -НАЧАЛЬНОЕ ЗНАЧЕНИЕ",ALO
70 INPUT"       -КОНЕЧНОЕ  ЗНАЧЕНИЕ",ALK
80 INPUT"       -ШАГ ИЗМЕНЕНИЯ",DAL
90 REM ПОДГОТОВКА "ГОЛОВКИ" ТАБЛИЦЫ
100 LPRINT"ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА"
110 LPRINT"ПРИ ДЛИНАХ СТОРОН";A;"И";B
120 LPRINT"УГОЛ МЕНЯЕТСЯ ОТ";ALO;"ДО";ALK
130 LPRINT"С ШАГОМ";DAL
140 GOSUB 500
150 LPRINT"НОМЕР","УГОЛ,РАД","ПЛОЩАДЬ"
160 GOSUB 500
170 N%=0
180 FOR ALFA=ALO TO ALK STEP DAL
190 N%=N%+1
200 S=FNS(ALFA)
210 LPRINT N%,ALFA,S
220 NEXT
230 GOSUB 500
240 END
500 REM ПОДПРОГРАММА "ЧЕРТА"
510 LPRINT"-----------------------------------------------"
520 RETURN
```

В строке 20 определена функция, зависящая от формального параметра X. В строке 200 указатель функции FNS(ALFA) включает в действие описанную в строке 20 функцию.

Для облегчения поиска в тексте программы подпрограммы "Черта" последняя начинается с номера строки 500.

Приведенная в примере 7.9 программа обеспечивает такой же экранный диалог и вывод, как и программа в примере 7.8.

Обращаем ваше внимание, что нецелесообразно использовать постоянные значения переменных A и B внутри цикла, поскольку это приводит к непроизводительной трате времени компьютера. Вычисление произведения A*B следует вынести за пределы цикла и использовать вспомогательную переменную, например W.

7.2.8. МАССИВЫ В ЦИКЛИЧЕСКИХ ПРОГРАММАХ

Что такое массив?

Массивом называется конечная упорядоченная совокупность значений, имеющих одинаковый тип. Каждому элементу массива присуще одинаковое имя. Местоположение каждого элемента в массиве определяется с помощью индексов, являющихся целыми числами без знака. В зависимости от структуры (числа измерений) массива различают массивы одно-, дву-, трехмерные и т. д. Для идентификации элементов в этих массивах используются соответственно один, два, три индекса и т. д. Для определения количества элементов массива необходимо знать его размерность (количество индексов) и диапазон изменения каждого индекса. В Бейсике принято использование массивов с размерностью, не превосходящей 255, причем наименьшее значение индекса равно нулю (иногда единице).

Вся необходимая для правильной работы программы информация о размерности массива, его объеме и типе элементов сосредотачивается в так называемом описателе массива. Например, описатель A(5) указывает, что элементы этого одномерного массива имеют вещественный тип

и индекс может меняться в диапазоне от 0 до 5. Таким образом, переменные A(0), A(1), A(2), A(3), A(4) и A(5) составляют данный массив.

Описатель массива P(1∅,5) показывает, что у элементов двумерного массива P первый индекс (номер строки) может принимать значения в интервале от 0 до 10, а второй индекс (номер столбца) — от 0 до 5. Таким образом, в памяти компьютера можно будет разместить не более 66 значений вещественного типа.

Для описания массивов в программе используется оператор DIM, имеющий вид

DIM список описателей массива

Например, оператор DIM A(5), P(1∅,5) обеспечивает возможность использования в программе двух массивов с указанными свойствами, а оператор DIM M% (15∅), H#(2,4,1∅) — массивов целых значений и значений удвоенной точности.

Важно подчеркнуть, что независимо от логической структуры массивов в памяти компьютера они всегда представляются в виде линейного списка. Так, размещение двумерного массива с описателем A(1,2)

A(0,0) A(0,1) A(0,2)
A(1,0) A(1,1) A(1,2)

будет организовано в памяти построчно: A(0,0), A(0,1), A(0,2), A(1,0), A(1,1), A(1,2).

Обратите внимание! Описатель массива по внешнему виду абсолютно тождественен переменной с числовыми индексами, поэтому смысл такой конструкции устанавливается по контексту.

Используем массивы в циклической программе

Модифицируйте последнюю версию программы так, чтобы воспользоваться массивами и переменными с индексами. В структуре новой программы следует сделать

определенные изменения. Так, принято, работая с массивами, сначала вычислить все элементы результирующих массивов и только после этого выводить их на печать.

В этой программе вам понадобятся два массива — один для хранения значений угла α и второй для хранения соответствующих значений площади. Для определения фактического количества элементов в каждом массиве воспользуйтесь формулой (7.3). Серия значений угла α получается в соответствии с выражением

$$\alpha_i = \alpha_0 + (i-1)\Delta\alpha, \; i = 1, 2, ..., N.$$

Ниже приведен текст программы, использующей массивы.

Пример 7.10.

```
10 REM ИСПОЛЬЗОВАНИЕ МАССИВА
20 DIM AL(15),S(15)
30 INPUT"ВВЕДИТЕ ДЛИНУ ПЕРВОЙ СТОРОНЫ",A
40 INPUT"ВВЕДИТЕ ДЛИНУ ВТОРОЙ СТОРОНЫ",B
50 PRINT"ВВЕДИТЕ ДЛЯ УГЛА АЛЬФА В РАДИАНАХ:"
60 INPUT"      -НАЧАЛЬНОЕ ЗНАЧЕНИЕ",ALO
70 INPUT"      -КОНЕЧНОЕ  ЗНАЧЕНИЕ",ALK
80 INPUT"      -ШАГ ИЗМЕНЕНИЯ",DAL
90 N%=(ALK-ALO)/DAL+1
100 W=A*B
110 FOR I=1 TO N%
120 AL(I)=ALO+(I-1)*DAL
130 S(I)=W*SIN(AL(I))
140 NEXT
150 REM  ВЫВОД РЕЗУЛЬТАТОВ
160 LPRINT"ВЫЧИСЛЕНИЕ ПЛОЩАДИ ПАРАЛЛЕЛОГРАММА"
170 LPRINT"ПРИ ДЛИНАХ СТОРОН";A;"И";B;". УГОЛ"
180 LPRINT"МЕНЯЕТСЯ ОТ";ALO;"ДО";ALK;"С ШАГОМ";DAL
190 GOSUB 250
200 LPRINT"НОМЕР","УГОЛ,РАД.","ПЛОЩАДЬ"
210 GOSUB 250
220 FOR I=1 TO N%:LPRINT I,AL(I),S(I):NEXT
230 GOSUB 250
240 END
250 REM ПОДПРОГРАММА "ЧЕРТА"
260 LPRINT"-----------------------------------------"
270 RETURN
```

В этой программе фактическое значение переменной N, обозначающей количество повторений цикла и количество элементов массивов AL и S, не должно превысить 15, так как именно такое максимальное значение индекса предусмотрено в описателях массива оператора DIM строки 20.

Циклические вычисления обеспечиваются операторами в строках 110—140. Вывод полученных результатов производится оператором цикла FOR, целиком размещенным в строке 220.

7.3. ИСПОЛЬЗУЕМ КОМАНДЫ БЕЙСИКА ДЛЯ РАБОТЫ С ПРОГРАММОЙ

Использование непрограммного режима

На стр. 323 описан первоначальный этап работы с компьютером — загрузка системы с Бейсиком. Получив на экране сообщение OK, вы можете начать свою работу.

Часто бывает очень удобно использовать непрограммный режим работы Бейсика. Например, вы хотите вычислить значение выражения

$$\sqrt{3^2+4^2}+5\cdot 2.$$

Для этого наберите на экране текст

```
PRINT SQR(3∧2+4∧2)+5*2.
```

После нажатия вами клавиши ET Бейсик немедленно интерпретирует этот оператор вывода, выполнит его, выведет на экран вычисленное значение и сообщит о своей готовности к дальнейшей работе:

```
15
OK
```

Автоматическая нумерация строк

Упрощения ввода составленной вами программы можно добиться применением команды AUTO, которая выполняет автоматическую генерацию номеров строк. После ввода очередной строки программы (после нажатия кла-

виши **ET**) в начале новой строки формируется очередной номер. Формат команды

AUTO [номер строки] [, шаг]

,

где номер строки указывает начальный номер первой строки программы, а шаг — шаг изменения номера строки. Если команда AUTO используется без параметров, то считается, что первый номер строки равен 10 и шаг равен 10.

Для выхода из режима автоматической генерации номеров строк следует одновременно нажать клавиши **CTRL** и **C**.

Исправление ошибок, редактирование программы

Во время набора строк на экране дисплея вами могут быть допущены ошибки. Пусть это обстоятельство не смущает вас.

●●● Проще всего исправить строку повторным ее набором под тем же номером. Например, в вашей программе имеется строка

```
7Ø PRINT A%, B%, C%
```

При вводе ее в память компьютера вы допустили ошибку и ввели эту строку в виде

```
7ØPRINT A%, B%, S%
```

Повторите набор этой же строки (но без ошибок, разумеется) с прежним номером:

```
7Ø PRINT A%, B%, C%
```

При этом старая строка с таким же номером будет уничтожена в памяти компьютера и заменена новой строкой с номером 70.

●●● При необходимости целиком удалить некоторую строку вашей программы следует набрать с новой строки номер удаляемой строки и нажать клавишу **ET** (без набора какого-либо текста).

●●● Более гибким средством редактирования является

Таблица 7.3

Операции редактирования

Назначение операции редактирования	Формат инструкции	Выполняемые действия
Перемещение курсора	Клавиша пробела (или ←)	Перемещение курсора на одну позицию вправо (или влево)
Вставка текста	I текст	Заданный текст вставляется в строку начиная с позиции курсора
	X текст	Заданный текст вставляется в конец строки. Если при этом превышается длина строки, то ввод блокируется
Стирание текста	[α] **D**	Справа от позиции курсора стираются α символов. Если α отсутствует, то стирается один символ
Замена текста	[α] **C** текст	Справа от позиции курсора α символов заменяются заданным текстом из α символов. Если α отсутствует, то заменяется один символ
Поиск символа	[α] **S** символ	Курсор останавливается перед заданным символом, встретившимся в строке α-й раз. Отсутствие α соответствует случаю $\alpha = 1$. Если заданный символ не найден, то курсор останавливается в конце строки
	[α] **K** символ	Отыскивается α-е появление заданного символа и стираются все символы, стоящие перед ним
Окончание редактирования	Клавиша **ET**	Заканчивается редактирование (выполнение команды EDIT) и новая строка передается в программу

использование режима редактирования, устанавливаемого командой EDIT, имеющей формат

EDIT номер строки

Эта команда вызывает загрузку указанной строки в специальный буфер памяти. Если вместо номера строки в команде стоит точка, то в буфер загружается строка, введенная последней.

После подачи команды EDIT вы можете воспользоваться операциями редактирования (табл. 7.3). Каждая операция задается соответствующей инструкцией — последовательностью нажатия определенных клавиш. При этом текст инструкции редактирования на экране не отображается.

Например, в проекте вашей программы имеется следующая строка:

28∅ IF B>2 THEN B=SIN(X)1))

Требуется исправить имя переменной с X на Y и одну из скобок аргумента указателя функции. Сначала воспользуйтесь командой EDIT 28∅. На экране в новой строке появится номер 280. Удерживая нажатой клавишу пробела получите на экране текст редактируемой строки.

Для замены двух подряд идущих символов X) на два символа Y(подведем курсор под первый исправляемый символ **X** с помощью клавиши ←:

28∅ IF B>2 THEN B=SIN(X)1))

Затем последовательным нажатием четырех клавиш введите инструкцию **2 C Y (**. Завершаем ввод инструкции нажатием клавиши **ESC**, если необходимо продолжить редактирование этой же строки, или клавиши **ET** при переходе к редактированию другой строки.

Перенумерация строк программы

В результате редактирования программы номера ее строк теряют регулярность. Вы можете заново перенуме-

ровать строки программы. При этом автоматически анализируются и модифицируются все имеющиеся в программе ссылки на изменившиеся номера строк (например, в операторе GOTO).

Перенумерация строк выполняется командой RENUM, имеющей формат

RENUM [новый номер строки] [, [старый номер строки][, новый шаг]]

Строкам программы начиная со старого номера строки присваиваются номера, которые начинаются с нового номера строки и отличаются друг от друга на новый шаг. Команда RENUM без параметров перенумеровывает строки начиная с первой. Новые номера начинаются с 10 и далее следуют с шагом 10. Эту команду удобно использовать для поиска неопределенных номеров строк.

Уничтожение строк программы

Стирание нескольких подряд идущих строк программы можно выполнить с помощью команды DELETE, имеющей формат, подобный формату команды LIST. Например, команда DELETE 50—190 стирает все строки с 50 по 190 включительно, команда DELETE-15∅ — все строки программы с начальной и до 150 включительно и т. д.

Вывод текста программы на экран

Окончив набор строк программы, вам следует убедиться в правильности ввода составленной программы. Это можно сделать с помощью команды LIST, имеющей формат

LIST [[номер первой строки] [— [номер последней строки]]]

Команда без параметров обеспечивает вывод на экран всего текста программы. При этом вывод строк может

быть прекращен с помощью клавиш **CTRL** и **C**. Нажатием клавиш **CTRL** и **S** вывод программы временно прерывается и может быть продолжен нажатием клавиш **CTRL** и **Q** с прерванного места.

Примеры:
LIST выводит все строки программы.
LIST 12Ø выводит строку с номером 120.
LIST 1ØØ— выводит все строки начиная с 100.
LIST—25Ø выводит все строки с начала до 250 включительно.
LIST 6Ø—27Ø выводит все строки с 60 по 270 включительно.

Вывод текста программы на принтер

Команда LLIST обеспечивает вывод текста программы на принтер. Формат этой команды и способы использования аналогичны команде LIST.

Запуск программы на выполнение

Для запуска программы воспользуйтесь командой RUN. Эта команда имеет две разновидности. В формате

RUN [номер строки]

команда вызывает запуск ранее загруженной программы с указанного в команде номера строки. Если номер строки в команде отсутствует, то программа запускается с первой имеющейся строки (со строки с наименьшим номером).

Во втором формате

RUN" имя файла"

команда сначала загружает в память программу из файла с указанным именем и затем запускает ее.

Запись программы на диск

Усталость или недостаток времени могут помешать вам завершить работу с вашей программой. Поэтому перед

окончанием работы с компьютером вы должны обеспечить сохранность вашей программы, переписав ее на диск в помощь команды SAVE. Команда имеет формат

$$\text{SAVE"имя файла"}\left[,\left\{\begin{matrix}A\\P\end{matrix}\right\}\right]$$

Здесь имя файла — допустимое в операционной системе SCP имя файла. Это имя берется в кавычки, хотя оно может быть задано и в виде символьного выражения. Расширение имени файла .BAS добавляется системой автоматически.

Если в дальнейшем сохраняемая программа будет включена в другую программу (с помощью команды MERGE) или будет обрабатываться компилятором Бейсика, то после слова SAVE задается параметр A. Сделать сохраняемую программу недоступной для других пользователей можно при помощи параметра P. При этом программа становится недоступной для внесения изменений и распечатки.

Если команда SAVE используется без параметров, то текст программы переносится и запоминается на диске в промежуточном коде, который вновь может быть загружен с помощью команды LOAD.

Например, по команде SAVE "PROG" программа, находящаяся в памяти компьютера, будет переписана на диск в файл под именем PROG.BAS.

Чтение программы с диска в память

С помощью команды LOAD можно переписать программу, хранящуюся на диске, в основную память компьютера и, если нужно, запустить ее для выполнения. Формат команды:

```
LOAD "имя файла" [, R]
```

Здесь имя файла — допустимое имя файла с расширением .BAS (как в формате команды SAVE). Команда LOAD проверяет расширение имени файла и загружает в основную память компьютера только содержимое файлов типа .BAS.

Если в команде LOAD используется параметр R, то загруженная программа сразу же запускается. Например, по команде LOAD "PROG", R в основную память компьютера загружается программа из файла с именем PROG.BAS и сразу же запускается для выполнения.

Переименование и уничтожение файлов

Работая с дисковыми файлами, вам может потребоваться переименовать файл, хранящийся на диске, или уничтожить файл на диске. С помощью команды NAME, имеющей формат

NAME"старое имя"AS"новое имя"

,

переименовывается дисковый файл. Допустим, что старое имя дискового файла PROG требуется заменить на новое имя SHET. Это можно сделать, воспользовавшись командой NAME "PROG" AS "SHET".

Уничтожение файла осуществляется с помощью команды KILL, имеющей формат

KILL "имя файла"

Например, команда KILL "PROG" ликвидирует файл с именем PROG.BAS, хранящийся на диске.

Вывод на экран справочника диска

Начиная работу с диском, вам может потребоваться информация о файлах, расположенных на этом диске.

С помощью команды FILES на экран может быть выведен список файлов. Формат команды:

FILES ["имя файла"]

При использовании команды без параметров на экран выводятся все файлы, имеющиеся на данном диске. Если использовать команду в виде FILES "*.BAS", то на экран будут выведены все имеющиеся файлы типа .BAS.

Завершение работы с Бейсиком

С помощью команды SYSTEM вы заканчиваете работу с интерпретатором Бейсика и выходите в операционную систему.

7.4. ОБЩАЯ СПРАВКА ПО БЕЙСИКУ

Обратившись к этому параграфу, вы можете систематизировать свои знания языка программирования Бейсик, более полно представить себе его возможности. При необходимости более углубленного изучения конструкций операторов и команд, не рассмотренных выше, рекомендуем воспользоваться литературными источниками или технической документацией версии языка Бейсик для компьютера Роботрон 1715.

Команды Бейсика

AUTO [номер строки] [, шаг] — автоматическое генерирование номеров строк программы

CLEAR — обнуление всех числовых переменных, символьные переменные делаются пустыми, открытые файлы закрываются

CONT — продолжение работы программы, прерванной нажатием клавиш **CTRL** и **C**, оператором STOP или END

DELETE [номер первой строки] [—] [номер послед-

ней строки] — уничтожение подряд идущих строк программы

EDIT номер строки — редактирование строки программы с помощью шести групп инструкций

FILES "имя файла" — вывод на экран сведений о файлах, хранящихся на диске

KILL "имя файла" — уничтожение любых файлов, хранящихся на диске

[L]LIST [[номер первой строки] [—[номер последней строки]]] — вывод на экран (принтер) программы или любой ее части

LOAD "имя файла" [,R] — загрузка программы с диска в основную память компьютера и, если нужно, запуск программы

MERGE "имя файла" — добавление к тексту, расположенному в основной памяти, текста, находящегося на диске

NAME "старое имя" AS "новое имя" — переименование файла

RENUM [новый номер строки [, старый номер строки [, новый шаг]]] — перенумерация строк программы, находящейся в основной памяти

RESET — закрытие файлов, запись на диск сведений о закрытых файлах

RUN $\left[\left\{\begin{matrix}\text{номер строки}\\ \text{"имя файла" [,R]}\end{matrix}\right\}\right]$ — запуск программы, возможно, после ее загрузки с диска в основную память

SAVE "имя файла" $\left[,\left\{\begin{matrix}A\\P\end{matrix}\right\}\right]$ — запись программы на диск из основной памяти

SYSTEM — выход из среды Бейсика в операционную систему

TRON (TROFF) — начать (закончить) работу программы трассировки TRACING

Операторы описания Бейсика

DATA список констант — организация набора данных для считывания оператором READ

DEFDBL список диапазонов букв — определение типа удвоенной точности для соответствующих имен программы

DEFINT список диапазонов букв — определение типа «целый» для соответствующих имен программы

DEFSTR список диапазонов букв — определение типа «символьный» для соответствующих имен программы

DEFISNG список диапазонов букв — определение типа «символьный» для соответствующих имен программы

DEFFN имя (список формальных параметров) = выражение — определение функции пользователя

DIM список описателей массива — определение массивов

ERASE список имен массива — очистка области памяти, занятой уже ненужными массивами

OPTION BASE $\begin{Bmatrix}0\\1\end{Bmatrix}$ — определение нижней границы индексов массивов программы

REM последовательность символов — комментарий в программе

RESTORE [номер строки] — начать считывание констант из набора данных оператора DATA, указанного номером строки

Операторы присваивания значений и операторы ввода-вывода Бейсика

INPUT [;] [символьная константа ;]список переменных — вывод на экран символьной константы и ввод значений для указанных переменных

[LET] переменная = выражение — оператор присваивания

LINE INPUT [;] [символьная константа ;] символьная переменная — вывод на экран символьной константы и ввод символа, набранного на клавиатуре

LPRINT (LPRINT USING) — реализуют вывод на принтер (см. операторы PRINT и PRINT USING)

PRINT [список выражений] — бесформатный вывод на экран

PRINT USING указатель формата; список выражений — форматный вывод на экран

READ список переменных — ввод данных из набора данных, организованного оператором DATA

WIDTH [LPRINT] длина — задание количества символов в строке вывода на экран (принтер)

WRITE [список выражений] — вывод на экран значений, разделяемых запятыми

Управляющие операторы Бейсика

END — завершение выполнения программы

FOR пар.цикла=нач.зн. TO кон.зн. [STEP шаг] — оператор организации детерминированного цикла

GOTO номер строки — безусловный переход

IF условие THEN операторы [ELSE операторы] — условный оператор

NEXT [параметр цикла] — конец тела цикла FOR—TO—STEP

ON числовое выражение GOTO список номеров строк — выбор одного из нескольких направлений перехода

STOP — прерывание выполнения программы

WEND — конец тела цикла WHILE

WHILE условие — оператор организации итерационного цикла

СПИСОК ЛИТЕРАТУРЫ

1. **Брябрин В. М.** Программное обеспечение персональных ЭВМ.— М.: Наука, 1988. — 272 с.

2. **Громов Г. Р.** Национальные информационные ресурсы: Проблемы промышленной эксплуатации. — М.: Наука, 1985.— 240 с.

3. **Дейтел Г.** Введение в операционные системы: В 2 т.: Пер. с англ. — М.: Мир, 1987.— 398 с.

4. **Дейт К.** Руководство по реляционной СУБД DB2: Пер. с англ.— М.: Финансы и статистика, 1988.— 230 с.

5. **Дойл У.** Табличный процессор Суперкалк для персонального компьютера: Пер. с англ.— М.: Финансы и статистика, 1987.— 320 с.

6. **Жигарев А. Н., Макарова Н. В., Путинцева М. А.** Основы компьютерной грамоты/Под ред. Н. В. Макаровой.— Л.: Машиностроение, Ленингр. отд-ние, 1987.— 255 с.

7. **Инструкция** по эксплуатации Robotron: Печатающее устройство. Техническая документация. 1986.— 65 с.

8. **Кочетков Г. Б.** Автоматизация конторского труда в США. — М.: Наука, 1986.— 160 с.

9. **Лори П.** Базы данных для микроЭВМ: Пер. с англ.— М.: Машиностроение, 1988.— 136 с.

10. **Мак-Кланг Кр. Дж., Герриери Дж. А., Мак-Кланг К. А. мл:** Микрокомпьютеры для юристов: Пер. с англ.— М.: Юридическая лит-ра, 1988.— 144 с.

11. **Мартин Дж.** Организация баз данных в вычислительных системах: Пер. с англ.— М.: Мир, 1980.— 664 с.

12. **Мейер Д.** Теория реляционных баз данных: Пер с англ.— М.: Мир, 1987.— 608 с.

13. **Нагао М., Катаяма Т., Уэмура С.** Структуры и базы данных.— М.: Мир, 1986. — 197 с.

14. **Персональные** компьютеры: Информатика для всех.— М.: Наука, 1987.— 149 с.

15. **Программное** обеспечение ПЭВМ Роботрон 1715: Система подготовки текстов Рефор. Описание применения.— Таллин: ТНУЦ, 1987.— 85 с.

16. **Программное** обеспечение ПЭВМ Роботрон 1715: Система про-

граммирования Бейсик. Описание применения.— Таллин: ТНУЦ, 1987.— 120 с.

17. **Программное** обеспечение ПЭВМ Роботрон 1715: Система управления реляционными базами данных Ребус. Описание применения.— Таллин: ТНУЦ, 1987.— 131 с.

18. **Пул Л.** Работа на персональном компьютере: Пер. с англ.— М.: Мир, 1986.— 383 с.

19. **Системное** руководство SCP: Руководство для оператора. Техническая документация: Роботрон. 1986.— 76 с.

20. **Уолш Б.** Программирование на Бейсике: Пер. с англ.— М.: Радио и связь, 1988.— 366 с.

21. **Шнайдер А.** Язык ассемблера для персонального компьютера фирмы IBM: Пер. с англ.— М.: Мир, 1988.— 406 с.

22. **Fertig R. F.** The software Revolution: Trends, Players, Market Dynamics in Personal Computer Software.— Noth Holland, 1985.— 304 p.

ОГЛАВЛЕНИЕ

ПРОИЗВОДСТВЕННОЕ ИЗДАНИЕ

Макарова Наталья Владимировна, **Докучаев** Александр Алексеевич, **Египко** Виктор Николаевич, **Семенов** Дмитрий Николаевич

РАБОТАЕМ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ РОБОТРОН 1715

Редактор *В. И. Баженко*
Переплет художника *Н. И. Абрамова*
Художественный редактор *С. С. Венедиктов*
Технический редактор *Т. М. Жилич*
Корректоры *И. Г. Иванова, Н. Б. Старостина*

ИБ № 6620

Сдано в набор 30.08.88. Подписано в печать 24.02.89. М—28910. Формат 70×108/32. Бумага типографская № 2. Гарнитура литературная. Печать высокая. Усл. печ. л. 16,1. Усл. кр.-отт. 16,1. Уч.-изд. л. 17,12. Доп. тир. 60 000 экз. Зак. 594. Цена 1 р. 30 к.

Ленинградское отделение ордена Трудового Красного Знамени издательства „Машиностроение", 191065, Ленинград, ул. Дзержинского, 10.

Сортавальская книжная типография Государственного комитета Карельской АССР по делам издательств, полиграфии и книжной торговли, 186750, Сортавала, ул. Карельская, 42.


РАБОТАЕМ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ